Theoretical Neuroscience

Theoretical Neuroscience

Computational and Mathematical Modeling of Neural Systems

Peter Dayan and L.F. Abbott

The MIT Press
Cambridge, Massachusetts
London, England

To our families

Contents

Series Foreword

Computational neuroscience is an approach to understanding the information content of neural signals by modeling the nervous system at many different structural scales, including the biophysical, the circuit, and the systems levels. Computer simulations of neurons and neural networks are complementary to traditional techniques in neuroscience. This book series welcomes contributions that link theoretical studies with experimental approaches to understanding information processing in the nervous system. Areas and topics of particular interest include biophysical mechanisms for computation in neurons, computer simulations of neural circuits, models of learning, representation of sensory information in neural networks, systems models of sensory-motor integration, and computational analysis of problems in biological sensing, motor control, and perception.

Terrence J. Sejnowski

Tomaso Poggio

Preface

Theoretical analysis and computational modeling are important tools for characterizing what nervous systems do, determining how they function, and understanding why they operate in particular ways. Neuroscience encompasses approaches ranging from molecular and cellular studies to human psychophysics and psychology. Theoretical neuroscience encourages crosstalk among these subdisciplines by constructing compact representations of what has been learned, building bridges between different levels of description, and identifying unifying concepts and principles. In this book, we present the basic methods used for these purposes and discuss examples in which theoretical approaches have yielded insight into nervous system function.

The questions what, how, and why are addressed by descriptive, mechanistic, and interpretive models, each of which we discuss in the following chapters. Descriptive models summarize large amounts of experimental data compactly yet accurately, thereby characterizing what neurons and neural circuits do. These models may be based loosely on biophysical, anatomical, and physiological findings, but their primary purpose is to describe phenomena, not to explain them. Mechanistic models, on the other hand, address the question of how nervous systems operate on the basis of known anatomy, physiology, and circuitry. Such models often form a bridge between descriptive models couched at different levels. Interpretive models use computational and information-theoretic principles to explore the behavioral and cognitive significance of various aspects of nervous system function, addressing the question of why nervous systems operate as they do.

descriptive models

mechanistic models

interpretive models

It is often difficult to identify the appropriate level of modeling for a particular problem. A frequent mistake is to assume that a more detailed model is necessarily superior. Because models act as bridges between levels of understanding, they must be detailed enough to make contact with the lower level yet simple enough to provide clear results at the higher level.

Organization and Approach

This book is organized into three parts on the basis of general themes. Part I, Neural Encoding and Decoding, (chapters 1–4) is devoted to the coding of information by action potentials and the representation of in-

Acknowledgments

We are extremely grateful to a large number of students at Brandeis, the Gatsby Computational Neuroscience Unit, and MIT, and colleagues at many institutions who have painstakingly read, commented on, and criticized numerous versions of all the chapters. We particularly thank Bard Ermentrout, Mark Goldman, John Hertz, Mark Kvale, Zhaoping Li, Eve Marder, and Read Montague for providing extensive discussion and advice on the entire book. A number of people read significant portions of the text and provided valuable comments, criticism, and insight: Bill Bialek, Pat Churchland, Nathaniel Daw, Dawei Dong, Peter Földiák, Fabrizio Gabbiani, Zoubin Ghahramani, Geoff Goodhill, David Heeger, Geoff Hinton, Ken Miller, Phil Nelson, Sacha Nelson, Bruno Olshausen, Mark Plumbley, Alex Pouget, Fred Rieke, John Rinzel, Emilio Salinas, Sebastian Seung, Mike Shadlen, Satinder Singh, Rich Sutton, Nick Swindale, Carl van Vreeswijk, Chris Williams, David Willshaw, Charlie Wilson, Angela Yu, and Rich Zemel.

We received significant additional assistance and advice from Greg DeAngelis, Andy Barto, Matt Beal, Sue Becker, Tony Bell, Paul Bressloff, Emery Brown, Matteo Carandini, Frances Chance, Yang Dan, Kenji Doya, Ed Erwin, John Fitzpatrick, David Foster, Marcus Frean, Ralph Freeman, Enrique Garibay, Frederico Girosi, Charlie Gross, Andreas Herz, Mike Jordan, Sham Kakade, Szabolcs Káli, Christof Koch, Simon Laughlin, John Lisman, Shawn Lockery, Guy Mayraz, Josh McDermott, Markus Meister, Earl Miller, Quaid Morris, Tony Movshon, Yuko Munakata, Randy O'Reilly, Simon Osindero, Tomaso Poggio, Clay Reid, Max Riesenhuber, Dario Ringach, Horacio Rotstein, Sam Roweis, Lana Rutherford, Ken Sugino, Alexei Samsonovich, Bob Shapley, Wolfram Schultz, Idan Segev, Terry Sejnowski, Jesper Sjöström, Haim Sompolinksy, Fiona Stevens, David Tank, Emo Todorov, Alessandro Treves, Gina Turrigiano, David Van Essen, Martin Wainwright, Xiao-Jing Wang, Chris Watkins, Max Welling, Jenny Whiting, Matt Wilson, Laurenz Wiskott, Danny Young, and Kechen Zhang. Thanks also to Quentin Huys, Philip Jonkers, Alexander Lerchner, Shih-Chii Liu, Máté Lengyel, Alex Loebel, Hadi Murr, Iain Murray, Jihwan Myung, John van Opstal, David Simon, Ed Snelson, and Rafael Yuste for pointing out errors in the text.

We thank Maneesh Sahani for advice and for indexing a substantial part of the text, Heidi Cartwright for creating the cover art, and Michael Rutter for his patience and consistent commitment. P.D. acknowledges the support of the Gatsby Charitable Foundation. Karen Abbott provided valuable help with the figures and with proofreading. Finally, we apologize to anyone we have inadvertently omitted from these lists.

I Neural Encoding and Decoding

1 Neural Encoding I: Firing Rates and Spike Statistics

1.1 Introduction

Neurons are remarkable among the cells of the body in their ability to propagate signals rapidly over large distances. They do this by generating characteristic electrical pulses called action potentials or, more simply, spikes that can travel down nerve fibers. Neurons represent and transmit information by firing sequences of spikes in various temporal patterns. The study of neural coding, which is the subject of the first four chapters of this book, involves measuring and characterizing how stimulus attributes, such as light or sound intensity, or motor actions, such as the direction of an arm movement, are represented by action potentials.

The link between stimulus and response can be studied from two opposite points of view. Neural encoding, the subject of chapters 1 and 2, refers to the map from stimulus to response. For example, we can catalog how neurons respond to a wide variety of stimuli, and then construct models that attempt to predict responses to other stimuli. Neural decoding refers to the reverse map, from response to stimulus, and the challenge is to reconstruct a stimulus, or certain aspects of that stimulus, from the spike sequences it evokes. Neural decoding is discussed in chapter 3. In chapter 4, we consider how the amount of information encoded by sequences of action potentials can be quantified and maximized. Before embarking on this tour of neural coding, we briefly review how neurons generate their responses and discuss how neural activity is recorded. The biophysical mechanisms underlying neural responses and action potential generation are treated in greater detail in chapters 5 and 6.

Properties of Neurons

Neurons are highly specialized for generating electrical signals in response to chemical and other inputs, and transmitting them to other cells. Some important morphological specializations, seen in figure 1.1, are the dendrites that receive inputs from other neurons and the axon that carries the neuronal output to other cells. The elaborate branching structure of

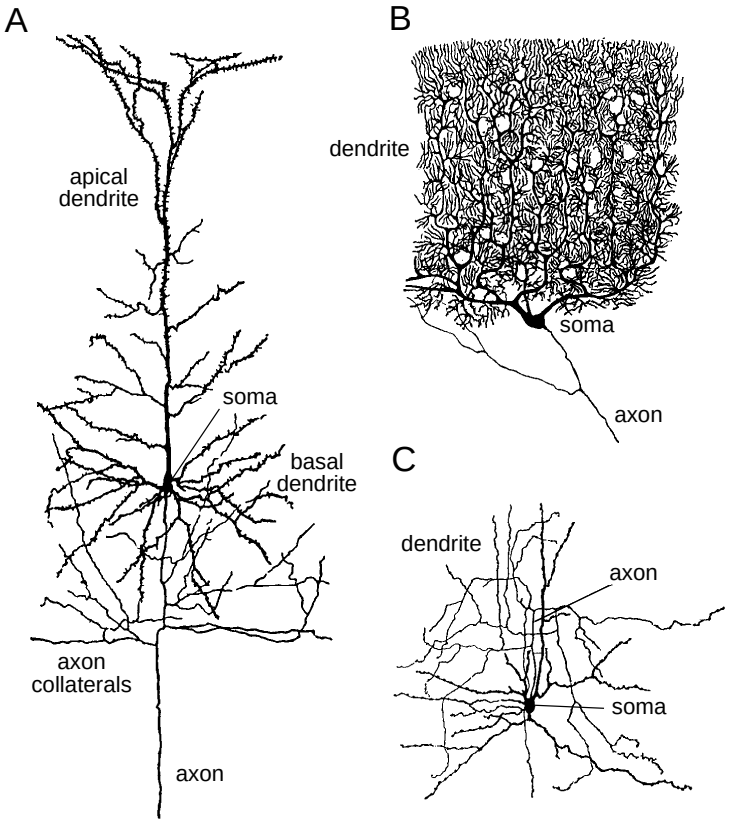

Figure 1.1 Diagrams of three neurons. (A) A cortical pyramidal cell. These are the primary excitatory neurons of the cerebral cortex. Pyramidal cell axons branch locally, sending axon collaterals to synapse with nearby neurons, and also project more distally to conduct signals to other parts of the brain and nervous system. (B) A Purkinje cell of the cerebellum. Purkinje cell axons transmit the output of the cerebellar cortex. (C) A stellate cell of the cerebral cortex. Stellate cells are one of a large class of interneurons that provide inhibitory input to the neurons of the cerebral cortex. These figures are magnified about 150-fold. (Drawings from Cajal, 1911; figure from Dowling, 1992.)

Action potentials are of great importance because they are the only form of membrane potential fluctuation that can propagate over large distances. Subthreshold potential fluctuations are severely attenuated over distances of 1 mm or less. Action potentials, on the other hand, are regenerated actively along axon processes and can travel rapidly over large distances without attenuation.

Axons terminate at synapses where the voltage transient of the action potential opens ion channels, producing an influx of Ca^{2+} that leads to the release of a neurotransmitter (figure 1.2B). The neurotransmitter binds to receptors at the signal-receiving or postsynaptic side of the synapse,

synapse

Weliky, M (2000) Correlated neuronal activity and visual cortical development. *Neuron* **27**:427–430.

Werblin, FS, & Dowling, JE (1969) Organization of the retina of the mudpuppy, *Necturus maculosus*. II. Intracellular recording. *Journal of Neurophysiology* **32**:339–355.

Wickens, J (1993) *A Theory of the Striatum*. New York: Pergamon Press.

Widrow, B, & Hoff, ME (1960) Adaptive switching circuits. *WESCON Convention Report* **4**:96–104.

Widrow, B, & Stearns, SD (1985) *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

Wiener, N (1958) *Nonlinear Problems in Random Theory*. New York: Wiley.

Williams, RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* **8**:229–256.

Wilson, HR, & Cowan, JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal* **12**:1–24.

Wilson, HR, & Cowan, JD (1973) A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* **13**:55–80.

Witkin, A (1983) Scale space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 2, 1019–1022. San Mateo, CA: Morgan Kaufmann.

Wörgötter, F, & Koch, C (1991) A detailed model of the primary visual pathway in the cat: Comparison of afferent excitatory and intracortical inhibitory connection schemes for orientation selectivity. *Journal of Neuroscience* **11**:1959–1979.

Yuste, R, & Sur, M (1999) Development and plasticity of the cerebral cortex: From molecules to maps. *Journal of Neurobiology* **41**:1–6.

Zador, A, Agmon-Snir, H, & Segev, I (1995) The morphoelectrotonic transform: A graphical approach to dendritic function. *Journal of Neuroscience* **15**:1669–1682.

Zemel, RS (1994) *A Minimum Description Length Framework for Unsupervised Learning*. Ph.D. dissertation, University of Toronto.

Zhang, K (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *Journal of Neuroscience* **16**:2112–2126.

Zhang, K, & Sejnowski, T (1999) Neural tuning: To sharpen or broaden? *Neural Computation* **11**:75–84.

Zhang, LI, Tao, HW, Holt, CE, Harris, WA, & Poo M-m (1998) A critical window for cooperation and competition among developing retinotectal synapses. *Nature* **395**:37–44.

Zigmond, MJ, Bloom, FE, Landis, SC, & Squire, LR, eds. (1998) *Fundamental Neuroscience*. San Diego: Academic Press.

Zipser, D, & Andersen, RA (1988) A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* **331**:679–684.

Zohary, E (1992) Population coding of visual stimuli by cortical neurons tuned to more than one dimension. *Biological Cybernetics* **66**:265–272.

Zucker, RS (1989). Short-term synaptic plasticity. *Annual Review of Neuroscience* **12**:13–31.

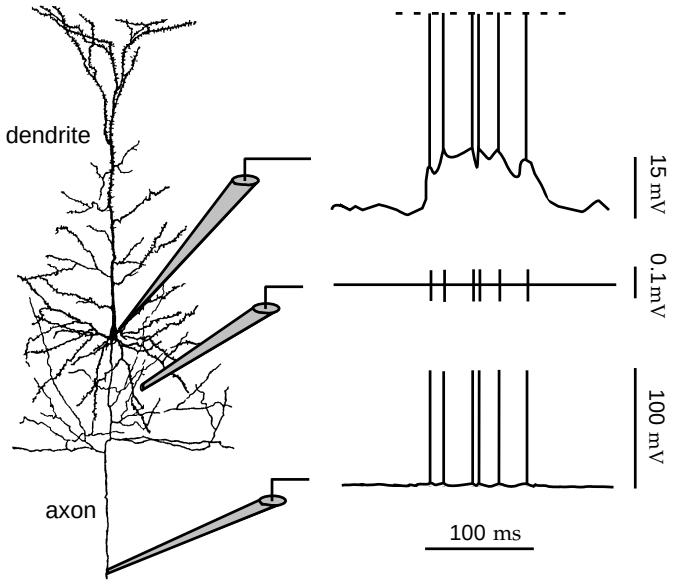

Figure 1.3 Three simulated recordings from a neuron. The top trace represents a recording from an intracellular electrode connected to the soma of the neuron. The height of the action potentials has been clipped to show the subthreshold membrane potential more clearly. The time scale is such that the action potential trajectory cannot be resolved. The bottom trace represents a recording from an intracellular electrode connected to the axon some distance away from the soma. The full height of the action potentials is indicated in this trace. The middle trace is a simulated extracellular recording. Action potentials appear as roughly equal positive and negative potential fluctuations with an amplitude of around 0.1 mV. This is roughly 1000 times smaller than the approximately 0.1 V amplitude of an intracellularly recorded action potential. (Neuron drawing is the same as figure 1.1A.)

are drawings, not real recordings; such intracellular axon recordings, although possible in some types of cells, are difficult and rare. Intracellular recordings from the soma are the norm, but intracellular dendritic recordings are increasingly being made as well. The subthreshold membrane potential waveform, apparent in the soma recording, is completely absent on the axon due to attenuation, but the action potential sequence in the two recordings is the same. This illustrates the important point that spikes, but not subthreshold potentials, propagate regeneratively down axons.

extracellular electrodes

The middle trace in figure 1.3 illustrates an idealized, noise-free extracellular recording. Here an electrode is placed near a neuron but it does not penetrate the cell membrane. Such recordings can reveal the action potentials fired by a neuron, but not its subthreshold membrane potentials. Extracellular recordings are typically used for in vivo experiments, especially those involving behaving animals. Intracellular recordings are sometimes made in vivo, but this is difficult to do. Intracellular recording is more commonly used for in vitro preparations, such as slices of neural tissue. The responses studied in this chapter are action potential sequences that can be recorded either intra- or extracellularly.

Solomon, RL, & Corbit, JD (1974) An opponent-process theory of motivation. I. Temporal dynamics of affect. *Psychological Review* **81**:119–145.

Somers, DC, Nelson, SB, & Sur, M (1995) An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience* **15**:5448–5465.

Sompolinsky, H, & Shapley, R (1997) New perspectives on the mechanisms for orientation selectivity. *Current Opinion in Neurobiology* **7**:514–522.

Song, S, Miller, KD, & Abbott, LF (2000) Competitive Hebbian learning through spike-timing dependent synaptic plasticity. *Nature Neuroscience* **3**:919–926.

Stemmler, M, & Koch, C (1999) How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nature Neuroscience* **2**:521–527.

Stevens, CM, & Zador, AM (1998) Novel integrate-and-fire-like model of repetitive firing in cortical neurons. In *Proceedings of the 5th Joint Symposium on Neural Computation*. La Jolla, CA: University of California at San Diego.

Strang, G (1976) *Linear Algebra and Its Applications*. New York: Academic Press.

Strong, SP, Koberle, R, de Ruyter van Steveninck, RR, & Bialek, W (1998) Entropy and information in neural spike trains. *Physical Review Letters* **80**:197–200.

Stuart, GJ, & Sakmann, B (1994) Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature* **367**:69–72.

Stuart, GJ, & Spruston, N (1998) Determinants of voltage attenuation in neocortical pyramidal neuron dendrites. *Journal of Neuroscience* **18**:3501–3510.

Sutton, RS (1988) Learning to predict by the methods of temporal difference. *Machine Learning* **3**:9–44.

Sutton, RS, & Barto, AG (1990) Time-derivative models of Pavlovian conditioning. In M Gabriel, & JW Moore, eds., *Learning and Computational Neuroscience*, 497–537. Cambridge, MA: MIT Press.

Sutton, RS, & Barto, AG (1998) *Reinforcement Learning*. Cambridge, MA: MIT Press.

Swindale, NV (1996) The development of topography in the visual cortex: A review of models. *Network: Computation in Neural Systems* **7**:161–247.

Theunissen, FE, & Miller, JP (1991) Representation of sensory information in the cricket cercal sensory system. II. Information theoretic calculation of system accuracy and optimal tuning-curve widths of four primary interneurons. *Journal of Neurophysiology* **66**:1690–1703.

Tipping, ME, & Bishop, CM (1999) Mixtures of probabilistic principal component analyzers. *Neural Computation* **11**:443–482.

Titterington, DM, Smith, AFM, & Makov, UE (1985) *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley.

Tootell, RB, Silverman, MS, Switkes, E, & De Valois, RL (1982) Deoxyglucose analysis of retinotopic organization in primate striate cortex. *Science* **218**:902–904.

Touretzky, DS, Redish, AD, & Wan, HS (1993) Neural representation of space using sinusoidal arrays. *Neural Computation* **5**:869–884.

start at time 0 and end at time T, so $0 \le t_i \le T$ for all i. The spike sequence can also be represented as a sum of infinitesimally narrow, idealized spikes in the form of Dirac δ functions (see the Mathematical Appendix),

$$\rho(t) = \sum_{i=1}^{n} \delta(t - t_i) \,. \tag{1.1}$$

We call $\rho(t)$ the neural response function and use it to re-express sums over spikes as integrals over time. For example, for any well-behaved function $h(t)$, we can write

neural response function $\rho(t)$

$$\sum_{i=1}^{n} h(t - t_i) = \int_{-\infty}^{\infty} d\tau \, h(\tau)\rho(t - \tau) \,, \tag{1.2}$$

where the integral is over the duration of the trial. The equality follows from the basic defining equation for a δ function,

δ function

$$\int d\tau \, \delta(t - \tau)h(\tau) = h(t) \,, \tag{1.3}$$

provided that the limits of the integral surround the point t (if they do not, the integral is 0).

Because the sequence of action potentials generated by a given stimulus varies from trial to trial, neuronal responses are typically treated statistically or probabilistically. For example, they may be characterized by firing rates, rather than as specific spike sequences. Unfortunately, the term "firing rate" is applied conventionally to a number of different quantities. The simplest of these is what we call the spike-count rate, which is obtained by counting the number of action potentials that appear during a trial and dividing by the duration of the trial. We denote the spike-count rate by r, where

spike-count rate r

$$r = \frac{n}{T} = \frac{1}{T} \int_0^T d\tau \, \rho(\tau) \,. \tag{1.4}$$

The second equality follows from the fact that $\int d\tau \, \rho(\tau) = n$ and indicates that the spike-count rate is the time average of the neural response function over the duration of the trial.

The spike-count rate can be determined from a single trial, but at the expense of losing all temporal resolution about variations in the neural response during the course of the trial. A time-dependent firing rate can be defined by counting spikes over short time intervals, but this can no longer be computed from a single trial. For example, we can define the firing rate at time t during a trial by counting all the spikes that occurred between times t and $t + \Delta t$, for some small interval Δt, and dividing this count by Δt. However, for small Δt, which allows for high temporal resolution, the result of the spike count on any given trial is apt to be either 0 or 1, giving only two possible firing-rate values. The solution to this problem is to average over multiple trials. Thus, we define the time-dependent firing rate

Roweis, S (1998) EM Algorithms for PCA and SPCA. In MI Jordan, M Kearns, & SA Solla, eds., *Advances in Neural Information Processing Systems, 10*, 626–632. Cambridge, MA: MIT Press.

Roweis, S, & Ghahramani, Z (1999) A unifying review of linear Gaussian models. *Neural Computation* **11**:305–345

Rubin, DB, & Thayer, DT (1982) EM algorithms for ML factor analysis. *Psychometrika* **47**:69–76.

de Ruyter van Steveninck, R, & Bialek, W (1988) Real-time performance of a movement-sensitive neuron in the blowfly visual system: Coding and information transfer in short spike sequences. *Proceedings of the Royal Society of London*. **B234**:379–414.

Sakmann, B, & Neher, E (1983) *Single Channel Recording*. New York: Plenum.

Salinas, E, & Abbott, LF (1994) Vector reconstruction from firing rates. *Journal of Computational Neuroscience* **1**:89–107.

Salinas, E, & Abbott, LF (1995) Transfer of coded information from sensory to motor networks. *Journal of Neuroscience* **15**:6461–6474.

Salinas, E, & Abbott, LF (1996). A model of multiplicative neural responses in parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America* **93**:11956–11961.

Salinas, E, & Abbott, LF (2000) Do simple cells in primary visual cortex form a tight frame? *Neural Computation* **12**:313–336.

Salzman, CA, Shadlen, MN, & Newsome, WT (1992) Microstimulation in visual area MT: Effects on directional discrimination performance. *Journal of Neuroscience* **12**:2331–2356.

Samsonovich, A, & McNaughton, BL (1997) Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience* **17**:5900–5920.

Sanger, TD (1994) Theoretical considerations for the analysis of population coding in motor cortex. *Neural Computation* **6**:29–37.

Sanger, TD (1996) Probability density estimation for the interpretation of neural population codes. *Journal of Neurophysiology* **76**:2790–2793.

Saul, AB, & Humphrey, AL (1990) Spatial and temporal properties of lagged and nonlagged cells in the cat lateral geniculate nucleus. *Journal of Neurophysiology* **68**:1190–1208.

Saul, LK, & Jordan, ML (2000) Attractor dynamics in feedforward neural networks. *Neural Computation* **12**:1313–1335.

Schultz, W (1998) Predictive reward signal of dopamine neurons. *Journal of Neurophysiology* **80**:1–27.

Schultz, W, Romo, R, Ljungberg, T, Mirenowicz, J, Hollerman, JR, & Dickinson, A (1995) Reward-related signals carried by dopamine neurons. In JC Houk, JL Davis, & DG Beiser, eds., *Models of Information Processing in the Basal Ganglia*, 233–248. Cambridge, MA: MIT Press.

Schwartz, EL (1977) Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics* **25**:181–194.

Sclar, G, & Freeman, R (1982) Orientation selectivity in cat's striate cortex is invariant with stimulus contrast. *Experimental Brain Research* **46**:457–461.

Scott, DW (1992) *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: Wiley.

In the same way that the response function $\rho(t)$ can be averaged across trials to give the firing rate $\mathsf{r}(t)$, the spike-count firing rate can be averaged over trials, yielding a quantity that we refer to as the average firing rate. This is denoted by $\langle r \rangle$ and is given by

average firing rate $\langle r \rangle$

$$\langle r \rangle = \frac{\langle n \rangle}{T} = \frac{1}{T}\int_0^T d\tau \, \langle \rho(\tau) \rangle = \frac{1}{T}\int_0^T dt \, \mathsf{r}(t) \, . \tag{1.7}$$

The first equality indicates that $\langle r \rangle$ is just the average number of spikes per trial divided by the trial duration. The third equality follows from the equivalence of the firing rate and the trial-averaged neural response function within integrals (equation 1.6). The average firing rate is equal to both the time average of $\mathsf{r}(t)$ and the trial average of the spike-count rate r. Of course, a spike-count rate and average firing rate can be defined by counting spikes over any time period, not necessarily the entire duration of a trial.

The term "firing rate" is commonly used for all three quantities, $\mathsf{r}(t)$, r, and $\langle r \rangle$. Whenever possible, we use the terms "firing rate", "spike-count rate", and "average firing rate" for $\mathsf{r}(t)$, r, and $\langle r \rangle$, respectively, but when this becomes too cumbersome, the different mathematical notations serve to distinguish them. In particular, we distinguish the spike-count rate r from the time-dependent firing rate $\mathsf{r}(t)$ by using a different font and by including the time argument in the latter expression (unless $\mathsf{r}(t)$ is independent of time). The difference between the fonts is rather subtle, but the context should make it clear which rate is being used.

Measuring Firing Rates

The firing rate $\mathsf{r}(t)$ cannot be determined exactly from the limited data available from a finite number of trials. In addition, there is no unique way to approximate $\mathsf{r}(t)$. A discussion of the different methods allows us to introduce the concept of a linear filter and kernel that will be used extensively in the following chapters. We illustrate these methods by extracting firing rates from a single trial, but more accurate results could be obtained by averaging over multiple trials.

Figure 1.4 compares a number of ways of approximating $\mathsf{r}(t)$ from a spike sequence. Figure 1.4A shows 3 s of the response of a neuron in the inferotemporal cortex recorded while a monkey watched a video. Neurons in the region of cortex where this recording was made are selective for complex visual images, including faces. A simple way of extracting an estimate of the firing rate from a spike train like this is to divide time into discrete bins of duration Δt, count the number of spikes within each bin, and divide by Δt. Figure 1.4B shows the approximate firing rate computed using this procedure with a bin size of 100 ms. Note that with this procedure, the quantity being computed is really the spike-count firing rate over the duration of the bin, and that the firing rate $\mathsf{r}(t)$ within a given bin is approximated by this spike-count rate.

Oram, MW, Földiák, P, Perrett, DI, & Sengpiel, F (1998) The "ideal homunculus": Decoding neural population signals. *Trends in Neuroscience* **21**:259–265.

Orban, GA (1984) *Neuronal Operations in the Visual Cortex*. Berlin: Springer-Verlag.

O'Reilly, RC (1996) Biologically plausible error-driven learning using local activation differences: The generalised recirculation algorithm. *Neural Computation* **8**:895–938.

Paradiso, MA (1988) A theory for the use of visual orientation information which exploits the columnar structure of striate cortex. *Biological Cybernetics* **58**:35–49.

Parker, AJ, & Newsome, WT (1998) Sense and the single neuron: Probing the physiology of perception. *Annual Review of Neuroscience* **21**:227–277.

Patlak, J (1991) Molecular kinetics of voltage-dependent Na^+channels. *Physiological Review* **71**:1047–1080.

Pearl, J (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Pearlmutter, BA, & Parra, LC (1996) A context-sensitive generalization of ICA. In S-I Amari, L Xu, L-W Chan, & I King, eds., *Proceedings of the International Conference on Neural Information Processing 1996*, 151–157. Singapore: Springer-Verlag.

Pece, AEC (1992) Redundancy reduction of a Gabor representation: A possible computational role for feedback from primary visual cortex to lateral geniculate nucleus. In I Aleksander, & J Taylor, eds., *Artificial Neural Networks, 2*, 865–868. Amsterdam: Elsevier.

Percival, DB, & Waldron, AT (1993) *Spectral Analysis for Physical Applications*. Cambridge: Cambridge University Press.

Piepenbrock, C, & Obermayer, K (1999) The role of lateral cortical competition in ocular dominance development. In MS Kearns, SA Solla, & DA Cohn, eds., *Advances in Neural Information Processing Systems, 11*. Cambridge, MA: MIT Press.

Plumbley, MD (1991) *On Information Theory and Unsupervised Neural Networks*. Cambridge University Engineering Department, Cambridge, technical report CUED/F-INFENG/TR.78.

Poggio, GF, & Talbot WH (1981) Mechanisms of static and dynamic stereopsis in foveal cortex of the rhesus monkey. *Journal of Physiology* **315**:469–492.

Poggio, T (1990) A theory of how the brain might work. *Cold Spring Harbor Symposium on Quantitative Biology* **55**:899–910.

Pollen, D, & Ronner, S (1982) Spatial computations performed by simple and complex cells in the visual cortex of the cat. *Vision Research* **22**:101–118.

Poor, HV (1994) *An Introduction to Signal Detection and Estimation, Second Edition*. New York: Springer-Verlag.

Pouget A, & Sejnowski TJ (1995) Spatial representations in the parietal cortex may use basis functions. In G Tesauro, DS Touretzky & TK Leen, eds., *Advances in Neural Information Processing Systems, 7*, 157–164. San Mateo, CA: Morgan Kaufmann.

Pouget, A, & Sejnowski, TJ (1997) Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience* **9**:222–237.

counting the number of spikes within the window at each location. The jagged curve in figure 1.4C shows the result of sliding a 100 ms wide window along the spike train. The firing rate approximated in this way can be expressed as the sum of a window function over the times t_i for $i = 1, 2, \ldots, n$ when the n spikes in a particular sequence occurred,

$$r_{\text{approx}}(t) = \sum_{i=1}^{n} w(t - t_i)\,, \tag{1.8}$$

where the window function is

$$w(t) = \begin{cases} 1/\Delta t & \text{if } -\Delta t/2 \le t < \Delta t/2 \\ 0 & \text{otherwise}\,. \end{cases} \tag{1.9}$$

Use of a sliding window avoids the arbitrariness of bin placement and produces a rate that might appear to have a better temporal resolution. However, it must be remembered that the rates obtained at times separated by less than one bin width are correlated because they involve some of the same spikes.

The sum in equation 1.8 can also be written as the integral of the window function times the neural response function (see equation 1.2):

$$r_{\text{approx}}(t) = \int_{-\infty}^{\infty} d\tau\, w(\tau)\rho(t - \tau)\,. \tag{1.10}$$

The integral in equation 1.10 is called a linear filter, and the window function w, also called the filter kernel, specifies how the neural response function evaluated at time $t - \tau$ contributes to the firing rate approximated at time t.

linear filter and kernel

The jagged appearance of the curve in figure 1.4C is caused by the discontinuous shape of the window function used. An approximate firing rate can be computed using virtually any window function $w(\tau)$ that goes to 0 outside a region near $\tau = 0$, provided that its time integral is equal to 1. For example, instead of the rectangular window function used in figure 1.4C, $w(\tau)$ can be a Gaussian:

$$w(\tau) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{\tau^2}{2\sigma_w^2}\right)\,. \tag{1.11}$$

In this case, σ_w controls the temporal resolution of the resulting rate, playing a role analogous to Δt. A continuous window function like the Gaussian used in equation 1.8 generates a firing-rate estimate that is a smooth function of time (figure 1.4D).

Both the rectangular and the Gaussian window functions approximate the firing rate at any time, using spikes fired both before and after that time. A postsynaptic neuron monitoring the spike train of a presynaptic cell has access only to spikes that have previously occurred. An approximation of the firing rate at time t that depends only on spikes fired before t can be calculated using a window function that vanishes when its argument

Marmarelis, PZ, & Marmarelis, VZ (1978) *Analysis of Physiological Systems: The White-Noise Approach*. New York: Plenum Press.

Marom, S, & Abbott, LF (1994) Modeling state-dependent inactivation of membrane currents. *Biophysical Journal* **67**:515–520.

Marr, D (1970) A theory for cerebral neocortex. *Proceedings of the Royal Society of London*. **B176**:161–234.

Mascagni, M, & Sherman, A (1998) Numerical methods for neuronal modeling. In C Koch, & I Segev, eds., *Methods in Neuronal Modeling: From Synapses to Networks*, 569–606. Cambridge, MA: MIT Press.

Mathews, J, & Walker, RL (1970) *Mathematical Methods of Physics*. New York: Benjamin.

Mauk, MD, & Donegan, NH (1997) A model of Pavlovian conditioning based on the synaptic organization of the cerebellum. *Learning and Memory* **4**:130–158.

McCormick, DA (1990) Membrane properties and neurotransmitter actions. In GM Shepherd, ed., *The Synaptic Organization of the Brain*. New York: Oxford University Press.

Mehta, MR, Barnes, CA, & McNaughton, BL (1997) Experience-dependent, asymmetric expansion of hippocampal place fields. *Proceedings of the National Academy of Sciences of the United States of America* **94**:8918–8921.

Mehta, MR, Quirk, MC, & Wilson, M (2000) Experience dependent asymmetric shape of hippocampal receptive fields. *Neuron* **25**:707–715.

Miller, KD (1994) A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between on- and off-center inputs. *Journal of Neuroscience* **14**:409–441.

Miller, KD (1996a) Receptive fields and maps in the visual cortex: Models of ocular dominance and orientation columns. In E Domany, JL van Hemmen, & K Schulten, eds., *Models of Neural Networks, Volume 3*, 55–78. New York: Springer-Verlag.

Miller, KD (1996b) Synaptic economics: competition and cooperation in synaptic plasticity. *Neuron* **17**:371–374.

Miller, KD, Keller, JB, & Stryker, MP (1989) Ocular dominance column development: Analysis and simulation. *Science* **245**:605–615.

Miller, KD, & MacKay, DJC (1994) The role of constraints in Hebbian learning. *Neural Computation* **6**:100–126.

Minai, AA, & Levy, WB (1993) Sequence learning in a single trial. *International Neural Network Society World Congress of Neural Networks II*. Portland, OR: International Neural Network Society, 505–508.

Minsky, M, & Papert, S (1969) *Perceptrons*. Cambridge, MA: MIT Press.

Mirenowicz, J, & Schultz, W (1994) Importance of unpredictability for reward responses in primate dopamine neurons. *Journal of Neurophysiology* **72**:1024–1027.

Montague, PR, Dayan, P, Person, C, & Sejnowski TJ (1995) Bee foraging in uncertain environments using predictive Hebbian learning. *Nature* **377**:725–728.

Montague, PR, Dayan, P, & Sejnowski, TJ (1996) A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience* **16**:1936–1947.

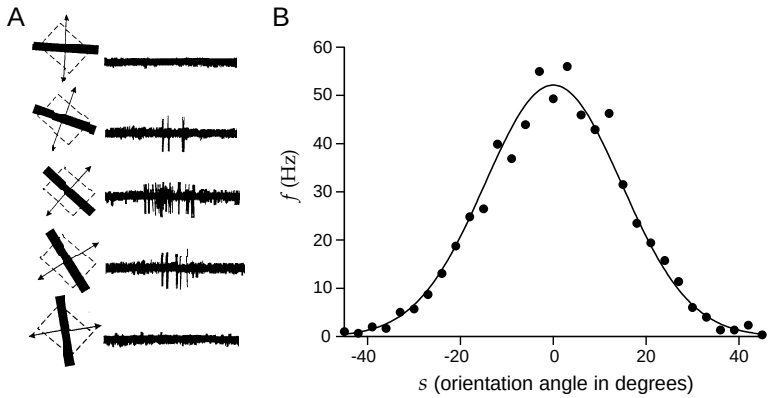

Figure 1.5 (A) Recordings from a neuron in the primary visual cortex of a monkey. A bar of light was moved across the receptive field of the cell at different angles. The diagrams to the left of each trace show the receptive field as a dashed square and the light source as a black bar. The bidirectional motion of the light bar is indicated by the arrows. The angle of the bar indicates the orientation of the light bar for the corresponding trace. (B) Average firing rate of a cat V1 neuron plotted as a function of the orientation angle of the light bar stimulus. The curve is a fit using the function 1.14 with parameters $r_{\max} = 52.14$ Hz, $s_{\max} = 0^\circ$, and $\sigma_f = 14.73^\circ$. (A adapted from Wandell, 1995, based on an original figure from Hubel and Wiesel, 1968; B data points from Henry et al., 1974).)

where s is the orientation angle of the light bar, $s_{\max}$ is the orientation angle evoking the maximum average response rate $r_{\max}$ (with $s - s_{\max}$ taken to lie in the range between -90° and $+90^\circ$), and σ_f determines the width of the tuning curve. The neuron responds most vigorously when a stimulus having $s = s_{\max}$ is presented, so we call $s_{\max}$ the preferred orientation angle of the neuron.

Response tuning curves can be used to characterize the selectivities of neurons in visual and other sensory areas to a variety of stimulus parameters. Tuning curves can also be measured for neurons in motor areas, in which case the average firing rate is expressed as a function of one or more parameters describing a motor action. Figure 1.6A shows an example of extracellular recordings from a neuron in primary motor cortex in a monkey that has been trained to reach in different directions. The stacked traces for each direction are rasters showing the results of five different trials. The horizontal axis in these traces represents time, and each mark indicates an action potential. The firing pattern of the cell, in particular the rate at which spikes are generated, is correlated with the direction of arm movement, and thus encodes information about this aspect of the motor action.

primary motor cortex M1

Figure 1.6B shows the response tuning curve of an M1 neuron plotted as a function of the direction of arm movement. Here the data points have been fitted by a tuning curve of the form

cosine tuning curve

$$f(s) = r_0 + (r_{\max} - r_0)\cos(s - s_{\max})\,, \tag{1.15}$$

where s is the reaching angle of the arm, $s_{\max}$ is the reaching angle associ-

Kalaska, JF, Caminiti, R, & Georgopoulos, AP (1983) Cortical mechanisms related to the direction of two-dimensional arm movements: Relations in parietal area 5 and comparison with motor cortex. *Experimental Brain Research* **51**:247–260.

Kandel, ER, & Schwartz, JH, eds. (1985) *Principles of Neural Science*. 2nd ed. New York: McGraw-Hill.

Kandel, ER, Schwartz, JH, & Jessel, TM, eds. (1991) *Principles of Neural Science*. 3rd ed. New York: McGraw-Hill.

Kandel, ER, Schwartz, JH, & Jessel, TM, eds. (2000) *Principles of Neural Science*. 4th ed. New York: McGraw-Hill.

Karasaridis, A, & Simoncelli, EP (1996) A filter design technique for steerable pyramid image transforms. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2387–2390. New York: IEEE.

Kawato, M, Hayakama, H, & Inui, T (1993) A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network: Computation in Neural Systems* **4**:415–422.

Kearns, MJ, & Vazirani, UV (1994) *An Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press.

Kehoe, EJ (1977) *Effects of Serial Compound Stimuli on Stimulus Selection in Classical Conditioning of the Rabbit Nictitating Membrane Response.* Ph.D. dissertation, University of Iowa.

Kempter R, Gerstner W, & van Hemmen JL (1999) Hebbian learning and spiking neurons. *Physical Review* **E59**:4498–4514.

Koch, C (1998) *Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press.

Koch, C, & Segev, I, eds. (1998) *Methods in Neuronal Modeling: From Synapses to Networks*. Cambridge, MA: MIT Press.

Konorski, J (1967) *Integrative Activity of the Brain*. Chicago: University of Chicago Press.

Lapicque, L (1907) Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et Pathologie Général* **9**:620–635.

Laughlin, S (1981) A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung* **36**:910–912.

Lauritzen, SL (1996) *Graphical Models*. Oxford: Clarendon Press.

Lee, C, Rohrer, WH, & Sparks, DL (1988) Population coding of saccadic eye movements by neurons in the superior colliculus. *Nature* **332**:357–360.

Leen, TK (1991) Dynamics of learning in recurrent feature-discovery networks. In RP Lippmann, JE Moody, & DS Touretzky, eds., *Advances in Neural Information Processing Systems, 3*. San Mateo, CA: Morgan Kaufmann, 70–76.

LeMasson, G, Marder, E, & Abbott, LF (1993) Activity-dependent regulation of conductances in model neurons. *Science* **259**:1915–1917.

Lewis, JE, & Kristan, WB (1998) A neuronal network for computing population vectors in the leech. *Nature* **391**:76–79.

Li, Z (1995) Modeling the sensory computations of the olfactory bulb. In E Domany, JL van Hemmen, & K Schulten, eds., *Models of Neural Networks, Volume 2*. New York: Springer-Verlag, 221–251.

Li, Z (1996) A theory of the visual motion coding in the primary visual cortex. *Neural Computation* **8**:705–730.

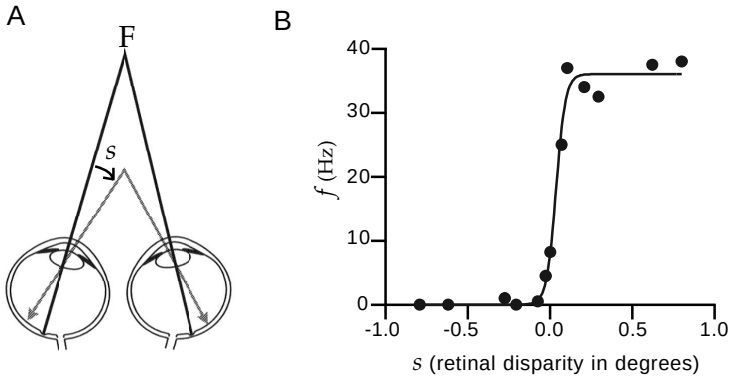

Figure 1.7 (A) Definition of retinal disparity. The gray lines with arrows show the location on each retina of an object located nearer than the fixation point F. The image from the fixation point falls at the fovea in each eye, the small pit where the black lines meet the retina. The image from a nearer object falls to the left of the fovea in the left eye and to the right of the fovea in the right eye. For objects farther away than the fixation point, this would be reversed. The disparity angle s is indicated in the figure. (B) Average firing rate of a cat V1 neuron responding to separate bars of light illuminating each eye, plotted as a function of the disparity. Because this neuron fires for positive s values, it is called a far-tuned cell. The curve is a fit using the function 1.17 with parameters $r_{\max} = 36.03$ Hz, $s_{1/2} = 0.036^\circ$, and $\Delta_s = 0.029^\circ$. (A adapted from Wandell, 1995; B data points from Poggio and Talbot, 1981.)

$\langle r \rangle = f(s)$ from trial to trial. While the map from stimulus to average response may be described deterministically, it is likely that single-trial responses such as spike-count rates can be modeled only in a probabilistic manner. For example, r values can be generated from a probability distribution with mean $f(s)$. The trial-to-trial deviation of r from $f(s)$ is considered to be noise, and such models are often called noise models. The standard deviation for the noise distribution either can be independent of $f(s)$, in which case the variability is called additive noise, or it can depend on $f(s)$. Multiplicative noise corresponds to having the standard deviation proportional to $f(s)$.

Response variability extends beyond the level of spike counts to the entire temporal pattern of action potentials. Later in this chapter, we discuss a model of the neuronal response that uses a stochastic spike generator to produce response variability. This approach takes a deterministic estimate of the firing rate, $r_{\text{est}}(t)$, and produces a stochastic spiking pattern from it. The spike generator produces variable numbers and patterns of action potentials, even if the same estimated firing rate is used on each trial.

1.3 What Makes a Neuron Fire?

Response tuning curves characterize the average response of a neuron to a given stimulus. We now consider the complementary procedure of av-

Heeger, DJ (1992) Normalization of cell responses in cat striate cortex. *Visual Neuroscience* **9**:181–198.

Heeger, DJ (1993) Modeling simple-cell direction selectivity with normalized, half-squared, linear operators. *Journal of Neurophysiology* **70**:1885–1898.

Henry, GH, Dreher, B, & Bishop, PO (1974) Orientation specificity of cells in cat striate cortex. *Journal of Neurophysiology* **37**:1394–1409.

Herrault, J, & Jutten, C (1986) Space or time adaptive signal processing by neural networks. In JS Denker, ed., *Neural Networks for Computing*. New York: American Institute for Physics.

Hertz, J, Krogh, A, & Palmer, RG (1991) *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.

Hille, B (1992) *Ionic Channels of Excitable Membranes*. Sunderland, MA: Sinauer Associates.

Hines, ML (1984) Efficient computation of branched nerve equations. *International Journal of Biomedical Computation* **15**:69–76.

Hines, ML, & Carnevale, NT (1997) The NEURON simulation environment. *Neural Computation* **9**:1179–1209.

Hinton, GE (1981) Shape representation in parallel systems. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1088–1096. Vancouver, BC.

Hinton, GE (1984) *Distributed Representations.* Technical report CMU-CS-84-157, Computer Science Department, Carnegie-Mellon University.

Hinton, GE (1989) Connectionist learning procedures. *Artificial Intelligence* **40**:185–234.

Hinton, GE (2000) *Training Products of Experts by Minimizing Contrastive Divergence.* Gatsby Computational Neuroscience Unit, University College London, technical report 2000-004.

Hinton, GE, Dayan, P, Frey, BJ, & Neal, RM (1995) The wake-sleep algorithm for unsupervised neural networks. *Science* **268**:1158–1160.

Hinton, GE, & Ghahramani, Z (1997) Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London*. **B352**:1177–1190.

Hinton, GE, & Sejnowski, TJ (1986) Learning and relearning in Boltzmann machines. In DE Rumelhart, & JL McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1, Foundations*. Cambridge, MA: MIT Press, 282-317.

Hinton, GE, & Zemel, RS (1994) Autoencoders, minimum description length and Helmholtz free energy. In JD Cowan, G Tesauro, & J Alspector, eds., *Advances in Neural Information Processing Systems, 6*, 3–10. San Mateo, CA: Morgan Kaufmann.

Hodgkin, AL, & Huxley, AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* **117**:500–544.

Holt, GR, Softky, GW, Koch, C, & Douglas, RJ (1996) Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *Journal of Neurophysiology* **75**:1806–1814.

Hopfield, JJ (1982) Neural networks and systems with emergent selective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* **79**:2554–2558.

taken by $s(t)$ during that trial, some of the mathematical analyses presented in this chapter and in chapter 2 are simplified if we define the stimulus at other times as well. It is convenient if integrals involving the stimulus are time-translationally invariant so that for any function h and time interval τ

$$\int_0^T dt\, h(s(t+\tau)) = \int_\tau^{T+\tau} dt\, h(s(t)) = \int_0^T dt\, h(s(t))\,. \tag{1.18}$$

periodic stimulus

To assure the last equality, we define the stimulus outside the time limits of the trial by the relation $s(T+\tau) = s(\tau)$ for any τ, thereby making the stimulus periodic.

The Spike-Triggered Average

The spike-triggered average stimulus, $C(\tau)$, is the average value of the stimulus a time interval τ before a spike is fired. In other words, for a spike occurring at time t_i, we determine $s(t_i - \tau)$, and then we sum over all n spikes in a trial, $i = 1, 2, \ldots, n$, and divide the total by n. In addition, we average over trials. Thus,

spike-triggered average $C(\tau)$

$$C(\tau) = \left\langle \frac{1}{n}\sum_{i=1}^{n} s(t_i - \tau) \right\rangle \approx \frac{1}{\langle n\rangle}\left\langle \sum_{i=1}^{n} s(t_i - \tau) \right\rangle . \tag{1.19}$$

The approximate equality of the last expression follows from the fact that if n is large, the total number of spikes on each trial is well approximated by the average number of spikes per trial, $n \approx \langle n\rangle$. We make use of this approximation because it allows us to relate the spike-triggered average to other quantities commonly used to characterize the relationship between stimulus and response (see below). Figure 1.8 provides a schematic description of the computation of the spike-triggered average. Each time a spike appears, the stimulus in a time window preceding the spike is recorded. Although the range of τ values in equation 1.19 is unlimited, the response is typically affected only by the stimulus in a window a few hundred milliseconds wide immediately preceding a spike. More precisely, we expect $C(\tau)$ to approach 0 for positive τ values larger than the correlation time between the stimulus and the response. If the stimulus has no temporal correlations with itself, we also expect $C(\tau)$ to be 0 for $\tau < 0$, because the response of a neuron cannot depend on future stimuli. In practice, the stimulus is recorded only over a finite time period, as indicated by the shaded areas in figure 1.8. The recorded stimuli for all spikes are then summed and the procedure is repeated over multiple trials.

The spike-triggered average stimulus can be expressed as an integral of the stimulus times the neural response function of equation 1.1. If we replace the sum over spikes with an integral, as in equation 1.2, and use the approximate expression for $C(\tau)$ in equation 1.19, we find

$$C(\tau) = \frac{1}{\langle n\rangle}\int_0^T dt\ \langle \rho(t)\rangle\, s(t-\tau) = \frac{1}{\langle n\rangle}\int_0^T dt\, r(t) s(t-\tau)\,. \tag{1.20}$$

Ferster, D (1994) Linearity of synaptic interactions in the assembly of receptive fields in cat visual cortex. *Current Opinion in Neurobiology* **4**:563–568.

Field, DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America* **A4**:2379–2394.

Field, DJ (1994) What is the goal of sensory coding? *Neural Computation* **6**:559–601.

Földiák, P (1989) Adaptive network for optimal linear feature extraction. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, 401–405. New York: IEEE Press.

Földiák, P (1991) Learning invariance from transformed sequences. *Neural Computation* **3**:194–200.

Foster, DJ, Morris, RGM, & Dayan, P (2000) Models of hippocampally dependent navigation using the temporal difference learning rule. *Hippocampus* **10**:1–16.

Freeman, WJ, & Schneider, W (1982) Changes in spatial patterns of rabbit olfactory EEG with conditioning to odors. *Psychophysiology* **19**:44–56.

Friston, KJ, Tononi, G, Reeke, GN Jr, Sporns, O, & Edelman, GM (1994) Value-dependent selection in the brain: Simulation in a synthetic neural model. *Neuroscience* **59**:229–243.

Fuster, JM (1995) *Memory in the Cerebral Cortex*. Cambridge, MA: MIT Press.

Gabbiani, F, & Koch, C (1998) Principles of spike train analysis. In C Koch, & I Segev, eds., *Methods of Neuronal Modeling*, 313–360. Cambridge, MA: MIT Press.

Gabbiani, F, Metzner, W, Wessel, R, & Koch, C (1996) From stimulus encoding to feature extraction in weakly electric fish. *Nature* **384**:564–567.

Gabor D (1946) Theory of communication. *Journal of the Institution of Electrical Engineers* **93**:429–457.

Gabriel, M, & Moore, JW, eds. (1990) *Learning and Computational Neuroscience*. Cambridge, MA: MIT Press.

Gallistel, CR (1990) *The Organization of Learning*. Cambridge, MA: MIT Press.

Gallistel, CR, & Gibbon, J (2000) Time, rate and conditioning. *Psychological Review* **107**:289–344.

Georgopoulos, AP, Kalaska, JF, Caminiti, R, & Massey, JT (1982) On the relations between the directions of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience* **2**:1527–1537.

Georgopoulos, AP, Kettner, RE, & Schwartz, AB (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *Neuroscience* **8**:2928–2937.

Georgopoulos, AP, Schwartz, AB, & Kettner, RE (1986) Neuronal population coding of movement direction. *Science* **243**:1416–1419.

Gershenfeld, NA (1999) *The Nature of Mathematical Modeling*. Cambridge: Cambridge University Press.

Gerstner, W (1998) Spiking neurons. In W Maass, & CM Bishop, eds., *Pulsed Neural Networks*. Cambridge, MA: MIT Press, 3–54.

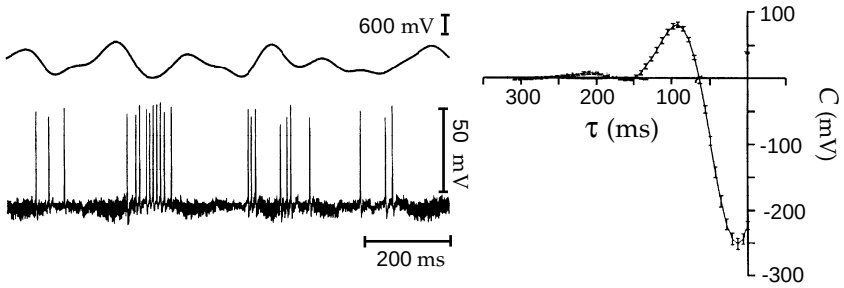

Figure 1.9 The spike-triggered average stimulus for a neuron of the electrosensory lateral-line lobe of the weakly electric fish *Eigenmannia*. The upper left trace is the potential used to generate the electric field to which this neuron is sensitive. The evoked spike train is plotted below the stimulus potential. The plot on the right is the spike-triggered average stimulus. (Adapted from Gabbiani et al., 1996.)

The spike-triggered average stimulus is widely used to study and characterize neural responses. Because $C(\tau)$ is the average value of the stimulus at a time τ before a spike, larger values of τ represent times farther in the past relative to the time of the triggering spike. For this reason, we plot spike-triggered averages with the time axis going backward compared to the normal convention. This allows the average spike-triggering stimulus to be read off from the plots in the usual left-to-right order.

Figure 1.9 shows the spike-triggered average stimulus for a neuron in the electrosensory lateral-line lobe of the weakly electric fish *Eigenmannia*. Weakly electric fish generate oscillating electric fields from an internal electric organ. Distortions in the electric field produced by nearby objects are detected by sensors spread over the skin of the fish. The lateral-line lobe acts as a relay station along the processing pathway for electrosensory signals. Fluctuating electrical potentials, such as that shown in the upper left trace of figure 1.9, elicit responses from electrosensory lateral-line lobe neurons, as seen in the lower left trace. The spike-triggered average stimulus, plotted at the right, indicates that, on average, the electric potential made a positive upswing followed by a large negative deviation prior to a spike being fired by this neuron.

The results obtained by spike-triggered averaging depend on the particular set of stimuli used during an experiment. How should this set be chosen? In chapter 2, we show that there are certain advantages to using a stimulus that is uncorrelated from one time to the next, a white-noise stimulus. A heuristic argument supporting the use of such stimuli is that in asking what makes a neuron fire, we may want to sample its responses to stimulus fluctuations at all frequencies with equal weight (i.e., equal power), and this is one of the properties of white-noise stimuli. In practice, white-noise stimuli can be generated with equal power only up to a finite frequency cutoff, but neurons respond to stimulus fluctuations only within a limited frequency range anyway. Figure 1.9 is based on such an approximate white-noise stimulus. The power in a signal as a function

Carpenter, GA, & Grossberg, S, eds. (1991) *Pattern Recognition by Self-Organizing Neural Network*. Cambridge, MA: MIT Press.

Chance, FS (2000) *Modeling Cortical Dynamics and the Responses of Neurons in the Primary Visual Cortex*. Ph.D. dissertation, Brandeis University.

Chance, FS, Nelson, SB, & Abbott, LF (1998) Synaptic depression and the temporal response characteristics of V1 simple cells. *Journal of Neuroscience* **18**:4785–4799.

Chance, FS, Nelson, SB, & Abbott, LF (1999) Complex cells as cortically amplified simple cells. *Nature Neuroscience* **2**:277–282.

Chauvin, Y, & Rumelhart, DE, eds. (1995) *Back Propagation: Theory, Architectures, and Applications*. Hillsdale, NJ: Erlbaum.

Churchland, PS, & Sejnowski, TJ (1992) *The Computational Brain*. Cambridge, MA: MIT Press.

Cohen, MA, & Grossberg, S (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics* **13**:815–826.

Compte, A, Brunel, N, Goldman-Rakic, PS, & Wang, XJ (2000) Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex* **10**:910–923.

Connor, JA, & Stevens, CF (1971) Prediction of repetitive firing behaviour from voltage clamp data on an isolated neurone soma. *Journal of Physiology* **213**:31–53.

Connor, JA, Walter, D, & McKown, R (1977) Neural repetitive firing: modifications of the Hodgkin-Huxley axon suggested by experimental results from crustacean axons. *Biophysical Journal* **18**:81–102.

Cover, TM (1965) Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition. *IEEE Transactions on Electronic Computers* **EC14**:326–334.

Cover, TM, & Thomas, JA (1991) *Elements of Information Theory*. New York: Wiley.

Cox, DR (1962) *Renewal Theory*. London: Methuen; New York: Wiley.

Cox, DR, & Hinckley, DV (1974) *Theoretical Statistics*. London: Chapman & Hall.

Cox, DR, & Isham, V (1980) *Point Processes*. New York: Chapman & Hall.

Crair, MC, Gillespie, DC, & Stryker, MP (1998) The role of visual experience in the development of columns in cat visual cortex. *Science* **279**:566–570.

Crowley, JC, & Katz, LC (1999) Development of ocular dominance columns in the absence of retinal input. *Nature Neuroscience* **2**:1125–1130.

Dan, Y, Atick, JJ, & Reid, RC (1996) Efficient coding of natural scenes in the lateral geniculate nucleus: Experimental test of a computational theory. *Journal of Neuroscience* **16**:3351–3362.

Daubechies, I (1992) *Ten Lectures on Wavelets*. Philadelphia: Society for Industrial and Applied Mathematics.

Daubechies, I, Grossmann, A, & Meyer, Y (1986) Painless nonorthogonal expansions. *Journal of Mathematical Physics* **27**:1271–1283.

Daugman, JG (1985) Uncertainty relation for resolution in space, spatial frequency, and orientation optimization by two-dimensional visual cortical filters. *Journal of the Optical Society of America* **2**:1160–1169.

Dayan, P, Hinton, GE, Neal, RM, & Zemel, RS (1995) The Helmholtz machine. *Neural Computation* **7**:889–904.

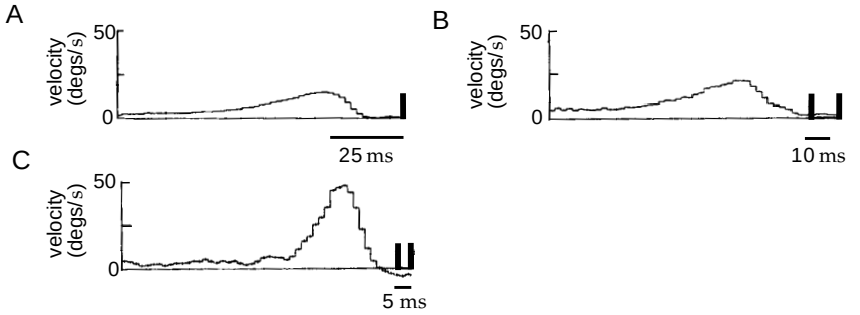

Figure 1.10 Single- and multiple-spike-triggered average stimuli for a blowfly H1 neuron responding to a moving visual image. (A) The average stimulus velocity triggered on a single spike. (B) The average stimulus velocity before two spikes with a separation of 10 ± 1 ms. (C) The average stimulus before two spikes with a separation of 5 ± 1 ms. (Data from de Ruyter van Steveninck and Bialek, 1988; figure adapted from Rieke et al., 1997.)

An approximation to white noise can be generated by choosing each s_m independently from a probability distribution with mean 0 and variance $\sigma_s^2/\Delta t$. Any reasonable probability function satisfying these two conditions can be used to generate the stimulus values within each time bin. A special class of white-noise stimuli, Gaussian white noise, results when the probability distribution used to generate the s_m values is a Gaussian function. The factor of $1/\Delta t$ in the variance indicates that the variability must be increased as the time bins get smaller. A number of other schemes for efficiently generating approximations of white-noise stimuli are discussed in the references at the end of this chapter.

Multiple-Spike-Triggered Averages and Spike-Triggered Correlations

In addition to triggering on single spikes, stimulus averages can be computed by triggering on various combinations of spikes. Figure 1.10 shows some examples of two-spike triggers. These results come from a study of the H1 movement-sensitive visual neuron of the blowfly. The H1 neuron detects the motion of visual images during flight in order to generate and guide stabilizing motor corrections. It responds to motion of the visual scene. In the experiments, the fly is held fixed while a visual image with a time-varying velocity $s(t)$ is presented. Figure 1.10A, showing the spike-triggered average stimulus, indicates that this neuron responds to positive angular velocities after a latency of about 15 ms. Figure 1.10B is the average stimulus prior to the appearance of two spikes separated by 10 ± 1 ms. In this case, the two-spike average is similar to the sum of two single-spike-triggered average stimuli displaced from one another by 10 ms. Thus, for 10 ms separations, two spikes occurring together tell us no more as a two-spike unit than they would individually. This result changes when shorter separations are considered. Figure 1.10C shows the

Bair, W, & Koch, C (1996) Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. *Neural Computation* **8**:1185–1202.

Bair, W, Koch, C, Newsome, WT, & Britten, KH (1994) Power spectrum analysis of bursting cells in area MT in the behaving monkey. *Journal of Neuroscience* **14**:2870–2892.

Baldi, P, & Heiligenberg, W (1988) How sensory maps could enhance resolution through ordered arrangements of broadly tuned receivers. *Biological Cybernetics* **59**:313–318.

Barlow, HB (1961) Possible principles underlying the transformation of sensory messages. In WA Rosenblith, ed., *Sensory Communication*. Cambridge, MA: MIT Press.

Barlow, HB (1989) Unsupervised learning. *Neural Computation* **1**:295–311.

Barlow, HB, & Levick, WR (1965) The mechanism of directionally selective units in the rabbit's retina. *Journal of Physiology* **193**:327–342.

Barto, AG, & Duff, M (1994) Monte Carlo matrix inversion and reinforcement learning. In G Tesauro, JD Cowan, & J Alspector, eds., *Advances in Neural Information Processing Systems, 6*, 598–605. San Mateo, CA: Morgan Kaufmann.

Barto, AG, Sutton, RS, & Anderson, CW (1983) Neuronlike elements that can solve difficult learning problems. *IEEE Transactions on Systems, Man, and Cybernetics* **13**:834–846.

Barto, AG, Sutton, RS, & Watkins, CJCH (1990) Learning and sequential decision making. In M Gabriel, & J Moore, eds., *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, 539–602. Cambridge, MA: MIT Press.

Battaglia, FP, & Treves, A (1998) Attractor neural networks storing multiple space representations: A model for hippocampal place fields. *Physical Review* **E58**:7738–7753.

Baum, LE, Petrie, E, Soules, G, & Weiss, N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics* **41**:164–171.

Bear, MF, Connors, BW, & Paradiso, MA (1996) *Neuroscience: Exploring the Brain*. Baltimore: Williams & Wilkins.

Becker, S, & Hinton, GE (1992) A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* **355**:161–163.

Becker, S, & Plumbley, M (1996) Unsupervised neural network learning procedures for feature extraction and classification. *International Journal of Applied Intelligence* **6**:185–203.

Bell, AJ, & Sejnowski, TJ (1995) An information maximisation approach to blind separation and blind deconvolution. *Neural Computation* **7**:1129–1159.

Bell, AJ, & Sejnowski, TJ (1996) Learning the higher-order structure of a natural sound. *Network: Computation in Neural Systems* **7**:261–267.

Bell, AJ, & Sejnowski, TJ (1997) The "independent components" of natural scenes are edge filters. *Vision Research* **37**:3327–3338.

Bellman, RE (1957) *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Ben-Yishai, R, Bar-Or, RL, & Sompolinsky, H (1995) Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences of the United States of America* **92**:3844–3848.

the presence of one spike may effect the occurrence of the other. If, however, the probability of generating an action potential is independent of the presence or timing of other spikes (i.e., if the spikes are statistically independent) the firing rate is all that is needed to compute the probabilities for all possible action potential sequences.

point process

A stochastic process that generates a sequence of events, such as action potentials, is called a point process. In general, the probability of an event occurring at any given time could depend on the entire history of preceding events. If this dependence extends only to the immediately preceding event, so that the intervals between successive events are independent, the point process is called a renewal process. If there is no dependence at all on preceding events, so that the events themselves are statistically independent, we have a Poisson process. The Poisson process provides an extremely useful approximation of stochastic neuronal firing. To make the presentation easier to follow, we separate two cases, the homogeneous Poisson process, for which the firing rate is constant over time, and the inhomogeneous Poisson process, which involves a time-dependent firing rate.

renewal process

Poisson process

The Homogeneous Poisson Process

We denote the firing rate for a homogeneous Poisson process by $r(t) = r$ because it is independent of time. When the firing rate is constant, the Poisson process generates every sequence of n spikes over a fixed time interval with equal probability. As a result, the probability $P[t_1, t_2, \ldots, t_n]$ can be expressed in terms of another probability function $P_T[n]$, which is the probability that an arbitrary sequence of exactly n spikes occurs within a trial of duration T. Assuming that the spike times are ordered so that $0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq T$, the relationship is

$$P[t_1, t_2, \ldots, t_n] = n! P_T[n] \left(\frac{\Delta t}{T}\right)^n . \tag{1.26}$$

This relationship is a special case of equation 1.37 derived below.

To compute $P_T[n]$, we divide the time T into M bins of size $\Delta t = T/M$. We can assume that Δt is small enough so that we never get two spikes within any one bin because, at the end of the calculation, we take the limit $\Delta t \to 0$. $P_T[n]$ is the product of three factors: the probability of generating n spikes within a specified set of the M bins, the probability of not generating spikes in the remaining $M - n$ bins, and a combinatorial factor equal to the number of ways of putting n spikes into M bins. The probability of a spike occurring in one specific bin is $r\Delta t$, and the probability of n spikes appearing in n specific bins is $(r\Delta t)^n$. Similarly, the probability of not having a spike in a given bin is $(1 - r\Delta t)$, so the probability of having the remaining $M - n$ bins without any spikes in them is $(1 - r\Delta t)^{M-n}$. Finally, the number of ways of putting n spikes into M bins is given by the

A.6 Annotated Bibliography

Most of the material in this appendix is covered in standard texts on mathematical methods such as **Mathews & Walker (1970)** and **Boas (1996)**. Discussion of relevant computational techniques, and code for implementing them, is available in **Press et al. (1992)**. Linear algebra is covered by **Strang (1976)**; linear and nonlinear differential equations, by **Jordan & Smith (1977)**; probability theory, by **Feller (1968)**; and Fourier transforms and the analysis of linear systems and electrical circuits, by **Siebert (1986)** and **Oppenheim & Willsky (1997)**. Mathematical approaches to biological problems are described in **Edelstein-Keshet (1988)** and **Murray (1993)**. Modern techniques of mathematical modeling are described by **Gershenfeld (1999)**.

General references for the other bodies of techniques used in the book include, for statistics, **Lindgren (1993)** and **Cox & Hinckley (1974)**, and for information theory, **Cover & Thomas (1991)**.

Thus the variance and mean of the spike count are equal. The ratio of these two quantities, $\sigma_n^2/\langle n\rangle$, is called the Fano factor and takes the value 1 for a homogeneous Poisson process, independent of the time interval T.

Fano factor

The probability density of time intervals between adjacent spikes is called the interspike interval distribution, and it is a useful statistic for characterizing spiking patterns. Suppose that a spike occurs at a time t_i for some value of i. The probability of a homogeneous Poisson process generating the next spike somewhere in the interval $t_i + \tau \le t_{i+1} < t_i + \tau + \Delta t$, for small Δt, is the probability that no spike is fired for a time τ, times the probability, $r\Delta t$, of generating a spike within the following small interval Δt. From equation 1.29, with $n = 0$, the probability of not firing a spike for period τ is $\exp(-r\tau)$, so the probability of an interspike interval falling between τ and $\tau + \Delta t$ is

interspike interval distribution

$$P[\tau \le t_{i+1} - t_i < \tau + \Delta t] = r\Delta t \exp(-r\tau)\,. \tag{1.31}$$

The probability density of interspike intervals is, by definition, this probability with the factor Δt removed. Thus, the interspike interval distribution for a homogeneous Poisson spike train is an exponential. The most likely interspike intervals are short ones, and long intervals have a probability that falls exponentially as a function of their duration.

From the interspike interval distribution of a homogeneous Poisson spike train, we can compute the mean interspike interval,

$$\langle\tau\rangle = \int_0^\infty d\tau\, \tau r \exp(-r\tau) = \frac{1}{r}\,, \tag{1.32}$$

and the variance of the interspike intervals,

$$\sigma_\tau^2 = \int_0^\infty d\tau\, \tau^2 r \exp(-r\tau) - \langle\tau\rangle^2 = \frac{1}{r^2}\,. \tag{1.33}$$

The ratio of the standard deviation to the mean is called the coefficient of variation,

coefficient of variation C_V

$$C_V = \frac{\sigma_\tau}{\langle\tau\rangle}\,, \tag{1.34}$$

and it takes the value 1 for a homogeneous Poisson process. This is a necessary, though not sufficient, condition to identify a Poisson spike train. Recall that the Fano factor for a Poisson process is also 1. For any renewal process, the Fano factor evaluated over long time intervals approaches the value C_V^2.

The Spike-Train Autocorrelation Function

The spike interval distribution measures the distribution of times between successive action potentials in a train. It is useful to generalize this concept and determine the distribution of times between any two spikes in

probability theory lie two objects: a sample space, Ω, and a measure. We begin by considering the simplest case of a finite sample space. Here, each element ω of the full sample space Ω can be thought of as one of the possible outcomes of a random process, for example, one of the 6^5 possible results of rolling five dice. The measure assigns a number γ_ω to each outcome ω, and these must satisfy $0 \leq \gamma_\omega \leq 1$ and $\sum_\omega \gamma_\omega = 1$.

sample space

probability measure

random variable

We are primarily interested in random variables (which are infamously neither random nor variable). A random variable is a mapping from a random outcome ω to a space such as the space of integers. An example is the number of ones that appear when five dice are rolled. Typically, a capital letter, such as S, is used for the random variable, and the corresponding lowercase letter, s in this case, is used for a particular value it might take. The probability that S takes the value s is then written as $P[S = s]$. In the text, we typically shorten this to $P[s]$, but here we keep the full notation (except in the following table). $P[S = s]$ is determined by the measures of the events for which $S = s$ and takes the value

$$P[S = s] = \sum_{\substack{\omega \text{ with} \\ S(\omega) = s}} \gamma_\omega \,. \tag{A.82}$$

The notation $S(\omega)$ refers to the value of S generated by the random event labeled by ω, and the sum is over all events for which $S(\omega) = s$.

Some key statistics for discrete random variables include the following.

Quantity	Definition	Alias
mean	$\langle s \rangle = \sum_s P[s]s$	$\overline{s}$, $\mathcal{E}[S]$
variance	$\mathrm{var}(S) = \langle s^2 \rangle - \langle s \rangle^2 = \sum_s P[s]s^2 - \langle s \rangle^2$	σ_s^2, $\mathcal{V}[S]$
covariance	$\langle s_1 s_2 \rangle - \langle s_1 \rangle \langle s_2 \rangle = \sum_{s_1 s_2} P[s_1, s_2]s_1 s_2 - \langle s_1 \rangle \langle s_2 \rangle$	$\mathrm{cov}(S_1, S_2)$

where S_1 and S_2 are two random variables defined over the same sample space. This links the two random variables, in that

$$P[S_1 = s_1, S_2 = s_2] = \sum_{\substack{\omega \text{ with} \\ S_1(\omega) = s_1, \\ S_2(\omega) = s_2}} \gamma_\omega \,, \tag{A.83}$$

and provides a basis for them to be correlated. Means are additive,

$$\langle s_1 + s_2 \rangle = \langle s_1 \rangle + \langle s_2 \rangle \,, \tag{A.84}$$

but other quantities typically are not, for example,

$$\mathrm{var}(S_1 + S_2) = \mathrm{var}(S_1) + \mathrm{var}(S_2) + 2\mathrm{cov}(S_1, S_2) \,. \tag{A.85}$$

independence

Two random variables are independent if $P[S_1 = s_1, S_2 = s_2] = P[S_1 = s_1]P[S_2 = s_2]$ for all s_1 and s_2. If S_1 and S_2 are independent, $\mathrm{cov}(S_1, S_2) = 0$, but the converse is not true in general.

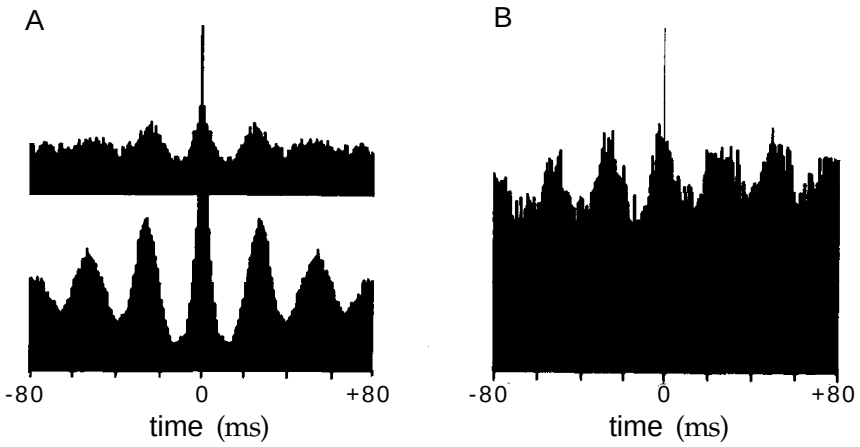

Figure 1.12 Autocorrelation and cross-correlation histograms for neurons in the primary visual cortex of a cat. (A) Autocorrelation histograms for neurons recorded in the right (upper) and left (lower) hemispheres show a periodic pattern indicating oscillations at about 40 Hz. The lower diagram indicates stronger oscillations in the left hemisphere. (B) The cross-correlation histogram for these two neurons shows that their oscillations are synchronized with little time delay. (Adapted from Engel et al., 1991.)

from each train. The spike-train autocorrelation function is an even function of τ, $Q_{\rho\rho}(\tau) = Q_{\rho\rho}(-\tau)$, but the cross-correlation function is not necessarily even. A peak at zero interval in a cross-correlation function signifies that the two neurons are firing synchronously. Asymmetric shifts in this peak away from 0 result from fixed delays between the firing of the two neurons, and they indicate nonsynchronous but phase-locked firing. Periodic structure in either an autocorrelation or a cross-correlation function or histogram indicates that the firing probability oscillates. Such periodic structure is seen in the histograms of figure 1.12, showing 40 Hz oscillations in neurons of cat primary visual cortex that are roughly synchronized between the two cerebral hemispheres.

The Inhomogeneous Poisson Process

When the firing rate depends on time, different sequences of n spikes occur with different probabilities, and $p[t_1, t_2, \ldots, t_n]$ depends on the spike times. Because spikes are still generated independently by an inhomogeneous Poisson process, their times enter into $p[t_1, t_2, \ldots, t_n]$ only through the time-dependent firing rate $\mathsf{r}(t)$. Assuming, as before, that the spike times are ordered $0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq T$, the probability density for n spike times (derived in appendix C) is

$$p[t_1, t_2, \ldots, t_n] = \exp\left(-\int_0^T dt\, \mathsf{r}(t)\right) \prod_{i=1}^{n} \mathsf{r}(t_i)\,. \tag{1.37}$$

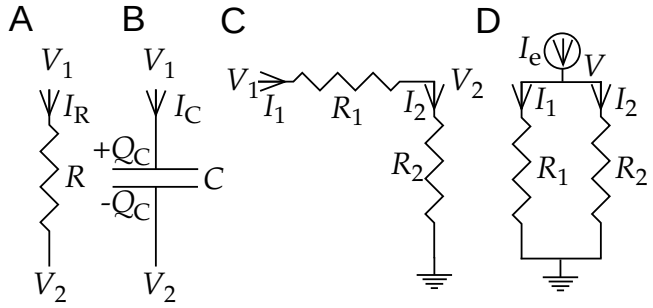

Figure A.1 Electrical circuit elements and resistor circuits. (A) Current I_R flows through a resistance R, producing a voltage drop $V_1 - V_2 = V_R$. (B) Charge $\pm Q_C$ is stored across a capacitance C, leading to a voltage $V_C = V_1 - V_2$ and a current I_C. (C) Series resistor circuit called a voltage divider. (D) Parallel resistor circuit. I_e represents an external current source. The lined triangle symbol at the bottom of the circuits in C and D represents an electrical ground, which is defined to be at 0 voltage.

change of charge, $I_C = dQ_C/dt$, to obtain the basic voltage-current relationship for a capacitor,

$$C\frac{dV_C}{dt} = I_C\,. \tag{A.76}$$

V-I relation for capacitor

Capacitance is measured in units of farads (F), defined as the capacitance for which 1 ampere of current causes a voltage change of 1 volt per second (1 F × 1 V/s = 1 A).

Kirchhoff's laws

The voltages at different points in a circuit and the currents flowing through various circuit elements can be computed using equations A.74 and A.76 and rules called Kirchhoff's laws. These state that voltage differences around any closed loop in a circuit must sum to 0, and that the sum of all the currents entering any point in a circuit must be 0. Applying the second of these rules to the circuit in figure A.1C, we find that $I_1 = I_2$. Ohm's law tells us that $V_1 - V_2 = I_1R_1$ and $V_2 = I_2R_2$. Solving these gives $V_1 = I_1(R_1 + R_2)$, which tells us that resistors arranged in series add, and $V_2 = V_1R_2/(R_1 + R_2)$, which is why this circuit is called a voltage divider.

In the circuit of figure A.1D, we have added an external source passing the current I_e. For this circuit, Kirchhoff's and Ohm's laws tells us that $I_e = I_1 + I_2 = V/R_1 + V/R_2$. This indicates how resistors add in parallel, $V = I_eR_1R_2/(R_1 + R_2)$.

Next, we consider the electrical circuit in figure A.2A, in which a resistor and capacitor are connected together. Kirchhoff's laws require that $I_C + I_R = 0$. Putting this together with equations A.74 and A.76, we find

$$C\frac{dV}{dt} = I_C = -I_R = -\frac{V}{R}\,. \tag{A.77}$$

Solving this gives

$$V(t) = V(0)\exp(-t/RC)\,, \tag{A.78}$$

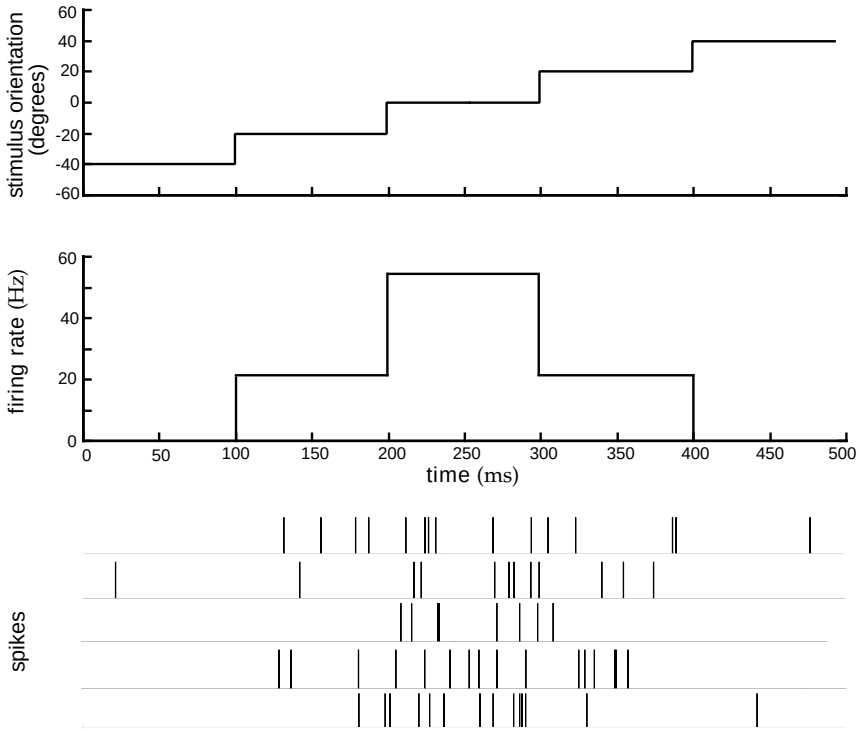

Figure 1.13 Model of an orientation-selective neuron. The orientation angle (top panel) was increased from an initial value of -40° by 20° every 100 ms. The firing rate (middle panel) was used to generate spikes (bottom panel) using a Poisson spike generator. The bottom panel shows spike sequences generated on five different trials.

Certain features of neuronal firing violate the independence assumption that forms the basis of the Poisson model, at least if a constant firing rate is used. We have already mentioned the absolute and relative refractory periods, which are periods of time following the generation of an action potential when the probability of a spike occurring is greatly or somewhat reduced. Frequently, these are most prominent features of real neuronal spike trains that are not captured by a Poisson model. Refractory effects can be incorporated into a Poisson model of spike generation by setting the firing rate to 0 immediately after a spike is fired, and then letting it return to its predicted value according to some dynamic rule such as an exponential recovery.

Comparison with Data

The Poisson process is simple and useful, but does it match data on neural response variability? To address this question, we examine Fano factors, interspike interval distributions, and coefficients of variation.

in conjugate pairs that combine to form a real function. Expression A.66 is not the correct solution if some of the eigenvalues are equal. The reader should consult the references for the solution in this case.

Equation A.66 determines how the evolution of $\mathbf{v}(t)$ in the neighborhood of $\mathbf{v}_\infty$ depends on the eigenvalues of $\mathbf{J}$. If we write $\lambda_\mu = \alpha_\mu + i\omega_\mu$,

$$\exp(\lambda_\mu t) = \exp(\alpha_\mu t)\left(\cos(\omega_\mu t) + i\sin(\omega_\mu t)\right). \tag{A.67}$$

This implies that modes with real eigenvalues ($\omega_\mu = 0$) evolve exponentially over time, and modes with complex eigenvalues ($\omega_\mu \neq 0$) oscillate with a frequency ω_μ. Recall that the eigenvalues are always real if $\mathbf{J}$ is a symmetric matrix. Modes with negative real eigenvalues ($\alpha_\mu < 0$ and $\omega_\mu = 0$) decay exponentially to 0, while those with positive real eigenvalues ($\alpha_\mu > 0$ and $\omega_\mu = 0$) grow exponentially. Similarly, the oscillations for modes with complex eigenvalues are damped exponentially to 0 if the real part of the eigenvalue is negative ($\alpha_\mu < 0$ and $\omega_\mu \neq 0$), and grow exponentially if the real part is positive ($\alpha_\mu > 0$ and $\omega_\mu \neq 0$).

attractor

unstable fixed point

marginal stability

Stability of the fixed point $\mathbf{v}_\infty$ requires the real parts of all the eigenvalues to be negative ($\alpha_\mu < 0$ for all μ). In this case, the point $\mathbf{v}_\infty$ is a stable fixed-point attractor of the system, meaning that $\mathbf{v}(t)$ will approach $\mathbf{v}_\infty$ if it starts from any point in the neighborhood of $\mathbf{v}_\infty$. If any real part is positive ($\alpha_\mu > 0$ for any μ), the fixed point is unstable. Almost any $\mathbf{v}(t)$ initially in the neighborhood of $\mathbf{v}_\infty$ will move away from that neighborhood. If $\mathbf{f}$ is linear, the exponential growth of $|\mathbf{v}(t) - \mathbf{v}_\infty|$ never stops in this case. For a nonlinear f, equation A.66 determines what happens only in the neighborhood of $\mathbf{v}_\infty$, and the system may ultimately find a stable attractor away from $\mathbf{v}_\infty$, either a fixed point, a limit cycle, or a chaotic attractor. In all these cases, the mode for which the real part of λ_μ takes the largest value dominates the dynamics as $t \to \infty$. If this real part is equal to 0, the fixed point is called marginally stable.

As mentioned previously, the analysis presented above as an approximation for nonlinear differential equations near a fixed point is exact if the original equation is linear. In the text, we frequently encounter linear equations of the form

$$\tau \frac{dv}{dt} = v_\infty - v\,. \tag{A.68}$$

This can be solved by setting $z = v - v_\infty$, rewriting the equation as $dz/z = -dt/\tau$, and integrating both sides:

$$\int_{z(0)}^{z(t)} dz' \frac{1}{z'} = \ln\left(\frac{z(t)}{z(0)}\right) = -\frac{t}{\tau}\,. \tag{A.69}$$

This gives $z(t) = z(0)\exp(-t/\tau)$ or

$$v(t) = v_\infty + (v(0) - v_\infty)\exp(-t/\tau)\,. \tag{A.70}$$

In some cases, we consider discrete rather than continuous dynamics defined over discrete steps $n = 1, 2, \ldots$ through a difference rather than a

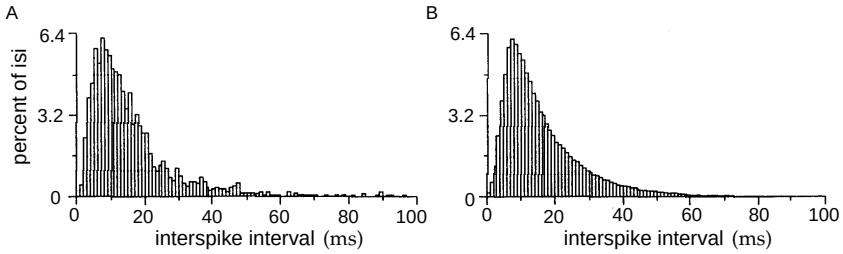

Figure 1.15 (A) Interspike interval distribution from an MT neuron responding to a moving, random-dot image. The probability of interspike intervals falling into the different bins, expressed as a percentage, is plotted against interspike interval. (B) Interspike interval histogram generated from a Poisson model with a stochastic refractory period. (Adapted from Bair et al., 1994.)

randomly moving dots with a variable amount of coherence imposed on their motion (see chapter 3 for a more detailed description). For interspike intervals longer than about 10 ms, the shape of this histogram is exponential, in agreement with equation 1.31. However, for shorter intervals there is a discrepancy. While the homogeneous Poisson distribution of equation 1.31 rises for short interspike intervals, the experimental results show a rapid decrease. This is the result of refractoriness making short interspike intervals less likely than the Poisson model would predict. Data on interspike intervals can be fitted more accurately by a gamma distribution,

gamma distribution

$$p[\tau] = \frac{r(r\tau)^k \exp(-r\tau)}{k!} \tag{1.39}$$

with $k > 0$, than by the exponential distribution of the Poisson model, which has $k = 0$.

Figure 1.15B shows a theoretical histogram obtained by adding a refractory period of variable duration to the Poisson model. Spiking was prohibited during the refractory period, and then was described once again by a homogeneous Poisson process. The refractory period was randomly chosen from a Gaussian distribution with a mean of 5 ms and a standard deviation of 2 ms (only random draws that generated positive refractory periods were included). The resulting interspike interval distribution of figure 1.15B agrees quite well with the data.

C_V values extracted from the spike trains of neurons recorded in monkeys from area MT and primary visual cortex (V1) are shown in figure 1.16. The data have been divided into groups based on the mean interspike interval, and the coefficient of variation is plotted as a function of this mean interval, equivalent to $1/\langle r \rangle$. Except for short mean interspike intervals, the values are near 1, although they tend to cluster slightly lower than 1, the Poisson value. The small C_V values for short interspike intervals are due to the refractory period. The solid curve is the prediction of a Poisson model with refractoriness.

The Poisson model with refractoriness provides a reasonably good description of a significant amount of data, especially considering its sim-

final equations. The clever idea of the Lagrange multiplier is to notice that the whole problem is symmetric with respect to the different components of $\Delta\mathbf{v}$. Choosing one c value, as we did above, breaks this symmetry and often complicates the algebra. To introduce the Lagrange multiplier, we simply define it as

$$\lambda = -\frac{[\nabla f]_c}{[\nabla g]_c}\,. \tag{A.57}$$

With this notation, the final set of equations (A.56) can be written as

$$[\nabla f]_a + \lambda[\nabla g]_a = 0\,. \tag{A.58}$$

Before, we had to say that these equations held only for $a \neq c$ because c was treated differently. Now, however, notice that the above equation when a is set to c is algebraically equivalent to the definition in equation A.57. Thus, we can say that equation A.58 applies for all a, and this provides a symmetric formulation of the problem of finding an extremum that often results in simpler algebra.

The final realization is that equation A.58 for all a is precisely what we would have derived if we had set out in the first place to find an extremum of the function $f(\mathbf{v}) + \lambda g(\mathbf{v})$ and forgotten about the constraint entirely. Of course this lunch is not completely free. From equation A.58, we derive a set of extremum points parameterized by the undetermined variable λ. To fix λ, we must substitute this family of solutions back into $g(\mathbf{v})$ and find the value of λ that satisfies the constraint that $g(\mathbf{v})$ equals the specified value. This provides the solution to the constrained problem.

A.3 Differential Equations

The most general differential equation we consider takes the form

$$\frac{d\mathbf{v}}{dt} = \mathbf{f}(\mathbf{v})\,, \tag{A.59}$$

where $\mathbf{v}$ is an N-component vector of time-dependent variables, and $\mathbf{f}$ is a vector of functions of $\mathbf{v}$. Unless it is unstable, allowing the absolute value of one or more of the components of $\mathbf{v}$ to grow without bound, this type of equation has three classes of solutions. For one class, called stable fixed points or point attractors, $\mathbf{v}(t)$ approaches a time-independent vector $\mathbf{v}_\infty$ ($\mathbf{v}(t) \to \mathbf{v}_\infty$) as $t \to \infty$. In a second class of solutions, called limit cycles, $\mathbf{v}(t)$ becomes periodic at large times and repeats itself indefinitely. For the third class of solutions, the chaotic ones, $\mathbf{v}(t)$ never repeats itself but the trajectory of the system lies in a limited subspace of the total space of allowed configurations called a strange attractor. Chaotic solutions are extremely sensitive to initial conditions.

fixed point

limit cycle

chaos

strange attractor

We focus most of our analysis on fixed point solutions, which are also called equilibrium points. For $\mathbf{v}_\infty$ to be a time-independent solution

equilibrium point

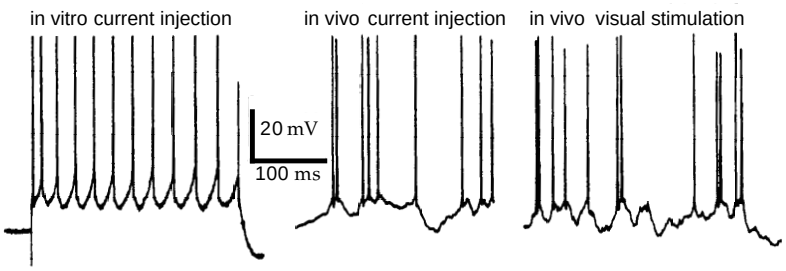

Figure 1.17 Intracellular recordings from cat V1 neurons. The left panel is the response of a neuron in an in vitro slice preparation to constant current injection. The center and right panels show recordings from neurons in vivo responding to either injected current (center) or a moving visual image (right). (Adapted from Holt et al., 1996.)

pendently of each other, or whether correlations between different spikes and different neurons carry significant amounts of information. We therefore contrast independent-spike and independent-neuron codes with correlation codes before addressing the issue of temporal coding.

Independent-Spike, Independent-Neuron, and Correlation Codes

The neural response, and its relation to the stimulus, are completely characterized by the probability distribution of spike times as a function of the stimulus. If spike generation can be described as an inhomogeneous Poisson process, this probability distribution can be computed from the time-dependent firing rate $r(t)$, using equation 1.37. In this case, $r(t)$ contains all the information about the stimulus that can be extracted from the spike train, and the neural code could reasonably be called a rate code. Unfortunately, this definition does not agree with common usage. Instead, we will call a code based solely on the time-dependent firing rate an independent-spike code. This refers to the fact that the generation of each spike is independent of all the other spikes in the train. If individual spikes do not encode independently of each other, we call the code a correlation code, because correlations between spike times may carry additional information. In reality, information is likely to be carried both by individual spikes and through correlations, and some arbitrary dividing line must be established to characterize the code. Identifying a correlation code should require that a significant amount of information be carried by correlations, for example, as much as is carried by the individual spikes.

independent-spike code

correlation code

A simple example of a correlation code would occur if significant amounts of information about a stimulus were carried by interspike intervals. In this case, if we considered spike times individually, independently of each other, we would miss the information carried by the intervals between them. This is just one example of a correlation code. Information could be carried by more complex relationships between spikes.

This equation implies a periodic continuation of f_n outside the range $0 \le n < N_t$, so that $f_{n+N_t} = f_n$ for all n. Consult the references in the bibliography for an analysis of the properties of the discrete Fourier transform and the quality of its approximation to the continuous Fourier transform. Note in particular that there is a difference between the discrete-time Fourier transform, which is the Fourier transform of a signal that is inherently discrete (i.e., is defined only at discrete points), and the discrete Fourier transform, given above, which is based on a finite number of samples of an underlying continuous function. If $f(t)$ is band-limited, meaning that $\tilde{f}(\omega) = 0$ for $|\omega| > \pi/\delta$, the sampling theorem states that $f(t)$ is completely determined by regular samples spaced at intervals $1/\delta$.

sampling theorem

Fourier transforms of functions of more than one variable involve a direct extension of the equations given above to multi-dimensional integrals. For example,

$$\tilde{f}(\omega_x, \omega_y) = \int dx \int dy\, f(x, y) \exp(i(\omega_x x + \omega_y y)). \tag{A.48}$$

The properties of multi-dimensional transforms are similar to those of one-dimensional transforms.

A.2 Finding Extrema and Lagrange Multipliers

An operation frequently encountered in the text is minimizing a quadratic form. In terms of vectors, this typically amounts to finding the matrix $\mathbf{W}$ that makes the product $\mathbf{W} \cdot \mathbf{v}$ closest to another vector $\mathbf{u}$ when averaged over a number of presentations of $\mathbf{v}$ and $\mathbf{u}$. The function to be minimized is the average squared error $\langle |\mathbf{W} \cdot \mathbf{v} - \mathbf{u}|^2 \rangle$, where the brackets denote averaging over all the different samples $\mathbf{v}$ and $\mathbf{u}$. Setting the derivative of this expression with respect to $\mathbf{W}$ (or equivalently its elements W_{ab}) to 0 gives the equation

minimization of quadratic form

$$\mathbf{W} \cdot \langle \mathbf{v}\mathbf{v} \rangle = \langle \mathbf{u}\mathbf{v} \rangle \quad \text{or} \quad \sum_{c=1}^{N} W_{ac} \langle v_c v_b \rangle = \langle u_a v_b \rangle \,. \tag{A.49}$$

Many variants of this equation, solved by a number of techniques, appear in the text.

Often, when a function $f(\mathbf{v})$ has to be minimized or maximized with respect to a vector $\mathbf{v}$, there is an additional constraint on $\mathbf{v}$ that requires another function $g(\mathbf{v})$ to be held constant. The standard way of dealing with this situation is to find the extrema of the function $f(\mathbf{v}) + \lambda g(\mathbf{v})$ where λ is a free parameter called a Lagrange multiplier. Once this is done, the value of λ is determined by requiring $g(\mathbf{v})$ to take the specified value. This procedure can appear a bit mysterious when first encountered, so we provide a rather extended discussion.

Lagrange multiplier

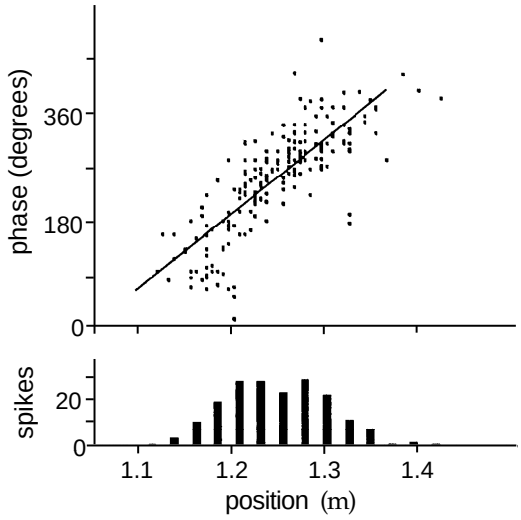

Figure 1.18 Position versus phase for a hippocampal place cell. Each dot in the upper figure shows the phase of the theta rhythm plotted against the position of the animal at the time when a spike was fired. The linear relation shows that information about position is contained in the relative phase of firing. The lower plot is a conventional place field tuning curve of spike count versus position. (Adapted from O'Keefe and Recce, 1993.)

place cell relative to the phase of the population theta rhythm gives additional information about the location of the rat not provided by place cells considered individually. The relationship between location and phase of place-cell firing shown in figure 1.18 means, for example, that we can distinguish two locations on opposite sides of the peak of a single neuron's tuning curve that correspond to the same firing rate, by knowing when the spikes occurred relative to the theta rhythm. However, the amount of additional information carried by correlations between place-field firing and the theta rhythm has not been fully quantified.

Temporal Codes

The concept of temporal coding arises when we consider how precisely we must measure spike times to extract most of the information from a neuronal response. This precision determines the temporal resolution of the neural code. A number of studies have found that this temporal resolution is on a millisecond time scale, indicating that precise spike timing is a significant element in neural encoding. Similarly, we can ask whether high-frequency firing-rate fluctuations carry significant information about a stimulus. When precise spike timing or high-frequency firing-rate fluctuations are found to carry information, the neural code is often identified as a temporal code.

Fourier Transforms

As outlined in the previous section, Fourier transforms provide a useful representation for functions when they are acted upon by translation-invariant linear operators.

Fourier transform

The Fourier transform of a function $f(t)$ is a complex function of a real argument ω given by

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} dt\, f(t) \exp(i\omega t)\,. \tag{A.35}$$

inverse Fourier transform

The Fourier transform $\tilde{f}(\omega)$ provides an alternative representation of the original function $f(t)$ because it can be inverted through

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega\, \tilde{f}(\omega) \exp(-i\omega t)\,. \tag{A.36}$$

This provides an inverse because

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \exp(-i\omega t) \int_{-\infty}^{\infty} dt'\, f(t') \exp(i\omega t') \tag{A.37}$$

$$= \int_{-\infty}^{\infty} dt'\, f(t') \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \exp(i\omega(t'-t)) = \int_{-\infty}^{\infty} dt'\, f(t')\delta(t'-t) = f(t)$$

by the definition of the δ function in equation A.30. The function $f(t)$ has to satisfy a set of criteria called the Dirichlet conditions for the inversion of the Fourier transform to be exact.

convolution

The convolution of two functions f and g is the integral

$$h(t) = \int_{-\infty}^{\infty} d\tau\, f(\tau) g(t-\tau)\,. \tag{A.38}$$

This is sometimes denoted by $h = f * g$. Note that the operation of multiplying a function by a linear filter and integrating, as in equation A.25, is a convolution. Fourier transforms are useful for dealing with convolutions because the Fourier transform of a convolution is the product of the Fourier transforms of the two functions being convolved,

$$\tilde{h}(\omega) = \tilde{f}(\omega)\tilde{g}(\omega)\,. \tag{A.39}$$

To show this, we note that

$$\tilde{h}(\omega) = \int_{-\infty}^{\infty} dt \exp(i\omega t) \int_{-\infty}^{\infty} d\tau\, f(\tau) g(t-\tau) \tag{A.40}$$

$$= \int_{-\infty}^{\infty} d\tau\, f(\tau) \exp(i\omega\tau) \int_{-\infty}^{\infty} dt\, g(t-\tau) \exp(i\omega(t-\tau))$$

$$= \int_{-\infty}^{\infty} d\tau\, f(\tau) \exp(i\omega\tau) \int_{-\infty}^{\infty} dt'\, g(t') \exp(i\omega t') \quad \text{where } t' = t-\tau\,,$$

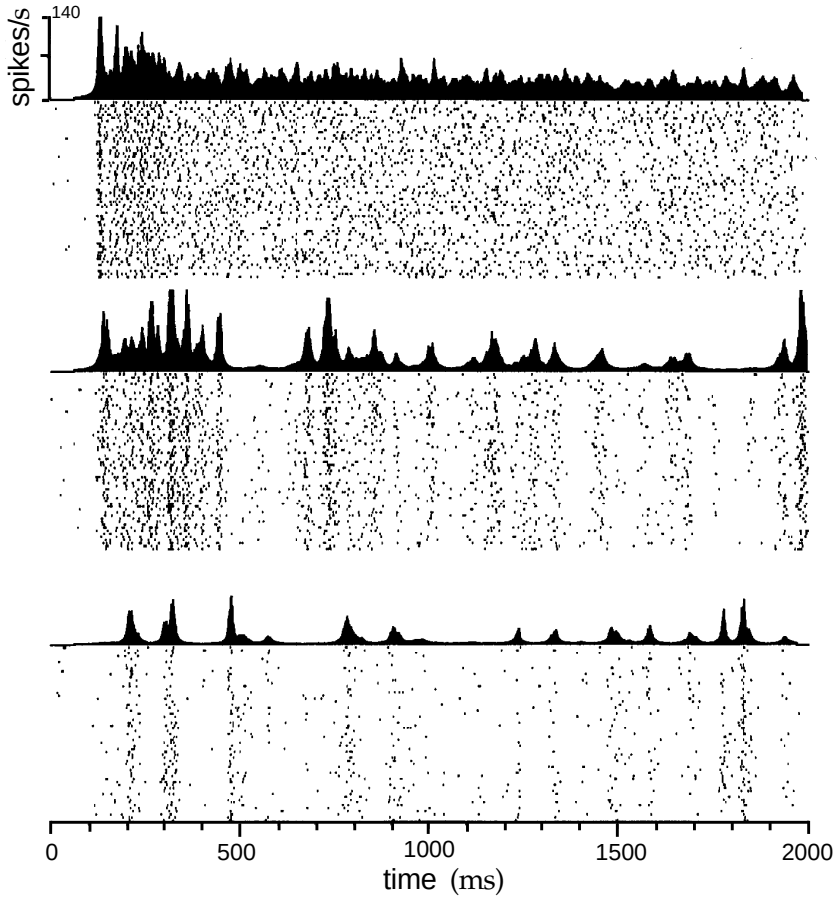

Figure 1.19 Time-dependent firing rates for different stimulus parameters. The rasters show multiple trials during which an MT neuron responded to the same moving, random-dot stimulus. Firing rates, shown above the raster plots, were constructed from the multiple trials by counting spikes within discrete time bins and averaging over trials. The three different results are from the same neuron but using different stimuli. The stimuli were always patterns of moving random dots, but the coherence of the motion was varied (see chapter 3 for more information about this stimulus). (Adapted from Bair and Koch, 1996.)

between the firing patterns of different neurons in a responding population and to understand their significance for neural coding.

1.6 Chapter Summary

With this chapter, we have begun our study of the way that neurons encode information using spikes. We used a sequence of δ functions, the neural response function, to represent a spike train and defined three types of firing rates: the time-dependent firing rate $r(t)$, the spike-count rate r,

linear integral operator

The analog of matrix multiplication for a function is the linear integral operator

$$\int_{-\infty}^{\infty} dt'\, W(t,t')v(t') \tag{A.22}$$

with the function values $W(t,t')$ playing the role of the matrix elements W_{ab}. The analog of the identity matrix is the Dirac δ function $\delta(t-t')$ discussed at the end of this section. The analog of a diagonal matrix is a function of two variables that is proportional to a δ function, $W(t,t') = h(t)\delta(t-t')$, for some function h.

functional inverse

All of the vector and matrix operations and properties defined above have functional analogs. Of particular importance are the functional inverse (which is not equivalent to an inverse function) that satisfies

$$\int_{-\infty}^{\infty} dt''\, W^{-1}(t,t'')W(t'',t') = \delta(t-t')\,, \tag{A.23}$$

translation invariance

and the analog of the Töplitz matrix, which is a linear integral operator that is translational-invariant, and thus can be written as

$$W(t,t') = K(t-t')\,. \tag{A.24}$$

linear filter

The linear integral operator then takes the form of a linear filter,

$$\int_{-\infty}^{\infty} dt'\, K(t-t')v(t') = \int_{-\infty}^{\infty} d\tau\, K(\tau)v(t-\tau)\,, \tag{A.25}$$

where we have made the replacement $t' \to t-\tau$.

The δ Function

Despite its name, the Dirac δ function is not a properly defined function, but rather the limit of a sequence of functions. In this limit, the δ function approaches 0 everywhere except where its argument is 0, and there it grows without bound. The infinite height and infinitesimal width of this function are matched so that its integral is 1. Thus,

$$\int dt\, \delta(t) = 1\,, \tag{A.26}$$

provided only that the limits of integration surround the point $t=0$ (otherwise the integral is 0). The integral of the product of a δ function with any continuous function f is

$$\int dt'\, \delta(t-t')f(t') = f(t) \tag{A.27}$$

for any value of t contained within the integration interval (if t is not within this interval, the integral is 0). These two identities normally provide enough information to use the δ function in calculations despite its unwieldy definition.

The first integral on the right side of the second equality is the complex conjugate of the Fourier transform of the stimulus,

$$\tilde{s}(\omega) = \frac{1}{T}\int_0^T d\tau\, s(\tau)\exp(i\omega\tau)\,. \tag{1.43}$$

The second integral, because of the periodicity of the integrand (when ω is an integer multiple of $2\pi/T$) is equal to $\tilde{s}(\omega)$. Therefore,

$$\tilde{Q}_{ss}(\omega) = |\tilde{s}(\omega)|^2\,, \tag{1.44}$$

which provides another definition of the stimulus power spectrum. It is the absolute square of the Fourier transform of the stimulus.

Although equations 1.40 and 1.44 are both sound, they do not provide a statistically efficient method of estimating the power spectrum of discrete approximations to white-noise sequences generated by the methods described in this chapter. That is, the apparently natural procedure of taking a white-noise sequence $s(m\Delta t)$ for $m = 1, 2, \ldots, T/\Delta t$, and computing the square amplitude of its Fourier transform at frequency ω,

$$\frac{\Delta T}{T}\left|\sum_{m=1}^{T/\Delta t} s(m\Delta t)\exp(-i\omega m\Delta t)\right|^2\,,$$

is a biased and extremely noisy way of estimating $\tilde{Q}_{ss}(\omega)$. This estimator is called the periodogram. The statistical problems with the periodogram, and some of the many suggested solutions, are discussed in almost any textbook on spectral analysis (see, e.g., Percival and Waldron, 1993).

periodogram

B: Moments of the Poisson Distribution

The average number of spikes generated by a Poisson process with constant rate r over a time T is

$$\langle n\rangle = \sum_{n=0}^{\infty} nP_T[n] = \sum_{n=0}^{\infty}\frac{n(\mathrm{r}T)^n}{n!}\exp(-\mathrm{r}T)\,, \tag{1.45}$$

and the variance in the spike count is

$$\sigma_n^2(T) = \sum_{n=0}^{\infty} n^2 P_T[n] - \langle n\rangle^2 = \sum_{n=0}^{\infty}\frac{n^2(\mathrm{r}T)^n}{n!}\exp(-\mathrm{r}T) - \langle n\rangle^2\,. \tag{1.46}$$

To compute these quantities, we need to calculate the two sums appearing in these equations. A good way to do this is to compute the moment-generating function

moment-generating function

$$g(\alpha) = \sum_{n=0}^{\infty}\frac{(\mathrm{r}T)^n\exp(\alpha n)}{n!}\exp(-\mathrm{r}T)\,. \tag{1.47}$$

where $f(a-b)$ is any function of the single variable $a-b$.

del operator ∇

For any real-valued function $E(\mathbf{v})$ of a vector $\mathbf{v}$, we can define the vector derivative (which is sometimes called del) of $E(\mathbf{v})$ as the vector $\nabla E(\mathbf{v})$ with components

$$[\nabla E(\mathbf{v})]_a = \frac{\partial E(\mathbf{v})}{\partial v_a}\,. \tag{A.10}$$

directional derivative

The derivative of $E(\mathbf{v})$ in the direction $\mathbf{u}$ is then

$$\lim_{\epsilon\to 0}\left(\frac{E(\mathbf{v}+\epsilon\mathbf{u})-E(\mathbf{v})}{\epsilon}\right) = \mathbf{u}\cdot\nabla E(\mathbf{v})\,. \tag{A.11}$$

Eigenvectors and Eigenvalues

eigenvector

An eigenvector of a square matrix $\mathbf{W}$ is a nonzero vector $\mathbf{e}$ that satisfies

$$\mathbf{W}\cdot\mathbf{e} = \lambda\mathbf{e} \tag{A.12}$$

eigenvalue

for some number λ called the eigenvalue. Possible values of λ are determined by solving the polynomial equation

$$\det(\mathbf{W}-\lambda\mathbf{I}) = 0\,. \tag{A.13}$$

Typically, but not always, this has N solutions if $\mathbf{W}$ is an N by N matrix, and these can be either real or complex. Complex eigenvalues come in complex-conjugate pairs if $\mathbf{W}$ has real-valued elements. We use the index μ to label the different eigenvalues and eigenvectors, λ_μ and $\mathbf{e}_\mu$. Note that μ identifies the eigenvector (and eigenvalue) to which we are referring; it does not signify a component of the eigenvector $\mathbf{e}_\mu$.

degeneracy

linear independence

If $\mathbf{e}$ is an eigenvector, $\alpha\mathbf{e}$ is also an eigenvector for any nonzero value of α. We can use this freedom to normalize eigenvectors so that $|\mathbf{e}| = 1$. If two eigenvectors, say $\mathbf{e}_1$ and $\mathbf{e}_2$, have the same eigenvalues $\lambda_1 = \lambda_2$, they are termed degenerate. Then, $\alpha\mathbf{e}_1 + \beta\mathbf{e}_2$ is also an eigenvector with the same eigenvalue, for any α and β that are not both 0. Apart from such degeneracies, an N by N matrix can have at most N eigenvectors, although some matrices have fewer. If $\mathbf{W}$ has N nondegenerate eigenvalues, the eigenvectors $\mathbf{e}_1,\ldots,\mathbf{e}_N$ are linearly independent, meaning that $\sum_\mu c_\mu\mathbf{e}_\mu = \mathbf{0}$ only if the coefficients $c_\mu = 0$ for all μ. These eigenvectors can be used to represent any N component vector $\mathbf{v}$ through the relation

$$\mathbf{v} = \sum_{\mu=1}^{N} c_\mu\mathbf{e}_\mu\,, \tag{A.14}$$

basis

with a unique set of coefficients c_μ. They are thus said to form a basis for the set of vectors $\mathbf{v}$.

symmetric matrix

The eigenvalues and eigenvectors of symmetric matrices (for which $\mathbf{W}^{\mathrm{T}} = \mathbf{W}$) have special properties, and for the remainder of this section, we con-

using the fact that the logarithm of a product is the sum of the logarithms of the multiplied terms. Using the approximation $\ln(1 - \mathsf{r}(t_i + m\Delta t)\Delta t) \approx -\mathsf{r}(t_i + m\Delta t)\Delta t$, valid for small Δt, we can simplify this to

$$\ln P[\text{no spikes}] = -\sum_{m=1}^{M} \mathsf{r}(t_i + m\Delta t)\Delta t\,. \tag{1.54}$$

In the limit $\Delta t \to 0$, the approximation becomes exact and this sum becomes the integral of $\mathsf{r}(t)$ from t_i to t_{i+1},

$$\ln P[\text{no spikes}] = -\int_{t_i}^{t_{i+1}} dt\,\mathsf{r}(t)\,. \tag{1.55}$$

Exponentiating this equation gives the result we need,

$$P[\text{no spikes}] = \exp\left(-\int_{t_i}^{t_{i+1}} dt\,\mathsf{r}(t)\right). \tag{1.56}$$

The probability density $p[t_1, t_2, \ldots, t_n]$ is the product of the densities for the individual spikes and the probabilities of not generating spikes during the interspike intervals, between time 0 and the first spike, and between the time of the last spike and the end of the trial period:

$$\begin{aligned} p[t_1, t_2, \ldots, t_n] &= \exp\left(-\int_0^{t_1} dt\,\mathsf{r}(t)\right)\exp\left(-\int_{t_n}^{T} dt\,\mathsf{r}(t)\right)\times \\ &\quad \mathsf{r}(t_n)\prod_{i=1}^{n-1}\mathsf{r}(t_i)\exp\left(-\int_{t_i}^{t_{i+1}} dt\,\mathsf{r}(t)\right). \end{aligned} \tag{1.57}$$

The exponentials in this expression all combine because the product of exponentials is the exponential of the sum, so the different integrals in this sum add up to form a single integral:

$$\begin{aligned} &\exp\left(-\int_0^{t_1} dt\,\mathsf{r}(t)\right)\exp\left(-\int_{t_n}^{T} dt\,\mathsf{r}(t)\right)\prod_{i=1}^{n-1}\exp\left(-\int_{t_i}^{t_{i+1}} dt\,\mathsf{r}(t)\right) \\ =\ &\exp\left(-\left(\int_0^{t_1} dt\,\mathsf{r}(t) + \sum_{i=1}^{n-1}\int_{t_i}^{t_{i+1}} dt\,\mathsf{r}(t) + \int_{t_n}^{T} dt\,\mathsf{r}(t)\right)\right) \\ =\ &\exp\left(-\int_0^{T} dt\,\mathsf{r}(t)\right). \end{aligned} \tag{1.58}$$

Substituting this into 1.57 gives the result in equation 1.37.

1.8 Annotated Bibliography

Braitenberg & Schuz (1991) provides some of the quantitative measures of neuroanatomical properties of cortex that we quote. **Rieke et al. (1997)**

$$\mathbf{v} \cdot \mathbf{u} = \sum_{a=1}^{N} v_a u_a \,. \tag{A.3}$$

matrix **W**

Matrix multiplication is a basic linear operation on vectors. An N_r by N_c matrix **W** is an array of N_r rows and N_c columns

$$\mathbf{W} = \begin{pmatrix} W_{11} & W_{12} & \dots & W_{1N_\mathrm{c}} \\ W_{21} & W_{22} & \dots & W_{2N_\mathrm{c}} \\ & & \vdots & \\ W_{N_\mathrm{r}1} & W_{N_\mathrm{r}2} & \dots & W_{N_\mathrm{r}N_\mathrm{c}} \end{pmatrix} \tag{A.4}$$

with elements W_{ab} for $a = 1, \dots, N_\mathrm{r}$ and $b = 1, \dots, N_\mathrm{c}$. In this text, the product of a matrix and a vector is written as $\mathbf{W} \cdot \mathbf{v}$. The dot implies multiplication and summation over a shared index, as it does for the dot product. If **W** is an N_r by N_c matrix and **v** is a N_c-component vector, $\mathbf{W} \cdot \mathbf{v}$ is an N_r-component vector with components

matrix-vector product

$$[\mathbf{W} \cdot \mathbf{v}]_a = \sum_{b=1}^{N_\mathrm{c}} W_{ab} v_b \,. \tag{A.5}$$

In conventional matrix notation, the product of a matrix and a vector is written as **Wv**, but we prefer to use the dot notation to avoid frequent occurrences of matrix transposes (see below). We similarly denote a matrix product as $\mathbf{W} \cdot \mathbf{M}$. Matrices can be multiplied in this way only if the number of columns of **W**, N_c, is equal to the number of rows of **M**. Then, $\mathbf{W} \cdot \mathbf{M}$ is a matrix with the same number of rows as **W** and the same number of columns as **M**, and with elements

matrix product

$$[\mathbf{W} \cdot \mathbf{M}]_{ab} = \sum_{c=1}^{N_\mathrm{c}} W_{ac} M_{cb} \,. \tag{A.6}$$

A vector, written as in equation A.1, is equivalent to a one-column, N-row matrix, and the rules for various matrix operations can thus be applied to vectors as well.

square matrix

Square matrices are those for which $N_\mathrm{r} = N_\mathrm{c} = N$. An important square matrix is the identity matrix **I** with elements

identity matrix

$$[\mathbf{I}]_{ab} = \delta_{ab} \,, \tag{A.7}$$

Kronecker delta

where the Kronecker delta is defined as

$$\delta_{ab} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \,. \end{cases} \tag{A.8}$$

diagonal matrix

Another important type of square matrix is the diagonal matrix, defined by

$$\mathbf{W} = \mathrm{diag}(h_1, h_2, \dots, h_N) = \begin{pmatrix} h_1 & 0 & \dots & 0 \\ 0 & h_2 & \dots & 0 \\ & & \vdots & \\ 0 & 0 & \dots & h_N \end{pmatrix} , \tag{A.9}$$

2 Neural Encoding II: Reverse Correlation and Visual Receptive Fields

2.1 Introduction

The spike-triggered average stimulus introduced in chapter 1 is a standard way of characterizing the selectivity of a neuron. In this chapter, we show how spike-triggered averages and reverse-correlation techniques can be used to construct estimates of firing rates evoked by arbitrary time-dependent stimuli. Firing rates calculated directly from reverse-correlation functions provide only a linear estimate of the response of a neuron, but in this chapter we also present various methods for including nonlinear effects such as firing thresholds.

retina
LGN
V1, area 17

Spike-triggered averages and reverse-correlation techniques have been used extensively to study properties of visually responsive neurons in the retina (retinal ganglion cells), lateral geniculate nucleus (LGN), and primary visual cortex (V1 or area 17). At these early stages of visual processing, the responses of some neurons (simple cells in primary visual cortex, for example) can be described quite accurately using this approach. Other neurons (complex cells in primary visual cortex, for example) can be described by extending the formalism. Reverse-correlation techniques have also been applied to responses of neurons in visual areas V2, area 18, and MT, but they generally fail to capture the more complex and nonlinear features typical of responses at later stages of the visual system. Descriptions of visual responses based on reverse correlation are approximate, and they do not explain how visual responses arise from the synaptic, cellular, and network properties of retinal, LGN, and cortical circuits. Nevertheless, they provide an important framework for characterizing response selectivities, a reference point for identifying and characterizing novel effects, and a basis for building mechanistic models, some of which are discussed at the end of this chapter and in chapter 7.

2.2 Estimating Firing Rates

In chapter 1, we discussed a simple model in which firing rates were estimated as instantaneous functions of the stimulus, using response tuning

duration of the trial, T,

$$E = \frac{1}{T}\int_0^T dt\ (\mathrm{r}_{\mathrm{est}}(t) - \mathrm{r}(t))^2\ . \tag{2.3}$$

This expression can be minimized by setting its derivative with respect to the function D to 0 (see appendix A). The result is that D satisfies an equation involving two quantities introduced in chapter 1, the firing rate-stimulus correlation function, $Q_{\mathrm{rs}}(\tau) = \int dt\, \mathrm{r}(t)\ s(t+\tau)/T$, and the stimulus autocorrelation function, $Q_{ss}(\tau) = \int dt\, s(t)\ s(t+\tau)/T$:

optimal kernel

$$\int_0^\infty d\tau'\, Q_{ss}(\tau - \tau')D(\tau') = Q_{\mathrm{rs}}(-\tau)\,. \tag{2.4}$$

The method we are describing is called reverse correlation because the firing rate–stimulus correlation function is evaluated at $-\tau$ in this equation.

Equation 2.4 can be solved most easily if the stimulus is white noise, although it can be solved in the general case as well (see appendix A). For a white-noise stimulus $Q_{ss}(\tau) = \sigma_s^2\delta(\tau)$ (see chapter 1), so the left side of equation 2.4 is

$$\sigma_s^2\int_0^\infty d\tau'\,\delta(\tau - \tau')D(\tau') = \sigma_s^2 D(\tau)\,. \tag{2.5}$$

As a result, the kernel that provides the best linear estimate of the firing rate is

white-noise kernel

$$D(\tau) = \frac{Q_{rs}(-\tau)}{\sigma_s^2} = \frac{\langle r\rangle C(\tau)}{\sigma_s^2}\,, \tag{2.6}$$

where $C(\tau)$ is the spike-triggered average stimulus and $\langle r\rangle$ is the average firing rate of the neuron. For the second equality, we have used the relation $Q_{\mathrm{rs}}(-\tau) = \langle r\rangle C(\tau)$ from chapter 1. Based on this result, the standard method used to determine the optimal kernel is to measure the spike-triggered average stimulus in response to a white-noise stimulus.

In chapter 1, we introduced the H1 neuron of the fly visual system, which responds to moving images. Figure 2.1 shows a prediction of the firing rate of this neuron obtained from a linear filter. The velocity of the moving image is plotted in 2.1A, and two typical responses are shown in 2.1B. The firing rate predicted from a linear estimator, as discussed above, and the firing rate computed from the data by binning and counting spikes are compared in figure 2.1C. The agreement is good in regions where the measured rate varies slowly, but the estimate fails to capture high-frequency fluctuations of the firing rate, presumably because of nonlinear effects not captured by the linear kernel. Some such effects can be described by a static nonlinear function, as discussed below. Others may require including higher-order terms in a Volterra or Wiener expansion.

10.7 Annotated Bibliography

The literature on unsupervised representational learning models is extensive. Recent reviews, from which we have borrowed include **Hinton (1989)**, **Bishop (1995)**, **Hinton & Ghahramani (1997)**, and **Becker & Plumbley (1996)**. These references also describe unsupervised learning methods such as IMAX (Becker & Hinton, 1992) that find statistical structure in the inputs directly rather than through causal models (see also projection pursuit, Huber, 1985). The field of belief networks or graphical statistical models (**Pearl, 1988**; Lauritzen, 1996; **Jordan, 1998**) provides an even more general framework for probabilistic generative models. Apart from **Barlow** (1961, **1989**), early inspiration for unsupervised learning models came from Uttley (1979) and Marr (1970), and from the adaptive resonance theory (ART) of Carpenter & Grossberg (1991).

Analysis by synthesis (e.g., Neisser, 1967), to which generative and recognition models are closely related, was developed in a statistical context by Grenander (1995), and was suggested by Mumford (1994) as a way of understanding hierarchical neural processing. Suggestions made in MacKay (1956), Pece (1992), Kawato et al. (1993), and Rao & Ballard (1997) can be seen in a similar light.

Nowlan (1991) introduced the mixtures of Gaussians architecture into neural networks. Mixture models are commonplace in statistics and are described by Titterington et al. (1985).

Factor analysis is described by Everitt (1984). Some of the differences and similarities between factor analysis and principal components analysis are brought out in Jolliffe (1986), Tipping & Bishop (1999), and Roweis & Ghahramani (1999). Rubin & Thayer (1982) discusses the use of EM for factor analysis. Roweis (1998) presents EM for principal components analysis.

Neal & Hinton (1998) describes $\mathcal{F}$ and its role in the EM algorithm (Baum et al., 1970; Dempster et al., 1977). EM is closely related to mean field methods in physics, as discussed in Jordan et al. (1998) and Saul & Jordan (2000). Hinton & Zemel (1994) and Zemel (1994) use $\mathcal{F}$ for unsupervised learning in a backpropagation network called the autoencoder, and these results are related to minimum description length coding (Risannen, 1989). Hinton et al. (1995) and Dayan et al. (1995) use $\mathcal{F}$ in the Helmholtz machine and the associated wake-sleep algorithm.

Olshausen & Field (1996) presents the sparse coding network based on Field's (1994) general analysis of sparse representations, and Olshausen (1996) develops some of the links to density estimation. Independent components analysis (ICA) was introduced as a problem by Herrault & Jutten (1986). The version of the ICA algorithm that we described is due to Bell & Sejnowski (1995) and Roth & Baram (1996), using the natural gradient trick of Amari (1999). The derivation we used is from MacKay (1996). Pearlmutter & Parra (1996) and Olshausen (1996) also derive maximum likelihood

the linear estimate of equation 2.1. At fixed stimulus energy, the integral in 2.1 measures the overlap between the actual stimulus and the most effective stimulus. In other words, it indicates how well the actual stimulus matches the most effective stimulus. Mismatches between these two reduce the value of the integral and result in lower predictions for the firing rate.

Static Nonlinearities

The optimal kernel produces an estimate of the firing rate that is a linear function of the stimulus. Neurons and nervous systems are nonlinear, so a linear estimate is only an approximation, albeit a useful one. The linear prediction has two obvious problems: there is nothing to prevent the predicted firing rate from becoming negative, and the predicted rate does not saturate, but instead increases without bound as the magnitude of the stimulus increases. One way to deal with these and some of the other deficiencies of a linear prediction is to write the firing rate as a background rate plus a nonlinear function of the linearly filtered stimulus. We use L to represent the linear term we have been discussing thus far:

$$L(t) = \int_0^\infty d\tau D(\tau) s(t-\tau)\,. \tag{2.7}$$

The modification is to replace the linear prediction $r_{est}(t) = r_0 + L(t)$ with the generalization

$r_{est}(t)$ with static nonlinearity

$$r_{est}(t) = r_0 + F(L(t))\,, \tag{2.8}$$

where F is an arbitrary function. F is called a static nonlinearity to stress that it is a function of the linear filter value evaluated instantaneously at the time of the rate estimation. If F is appropriately bounded from above and below, the estimated firing rate will never be negative or unrealistically large.

F can be extracted from data by means of the graphical procedure illustrated in figure 2.2A. First, a linear estimate of the firing rate is computed using the optimal kernel defined by equation 2.4. Next a plot is made of the pairs of points $(L(t), r(t))$ at various times and for various stimuli, where $r(t)$ is the actual rate extracted from the data. There will be a certain amount of scatter in this plot due to the inaccuracy of the estimation. If the scatter is not too large, however, the points should fall along a curve, and this curve is a plot of the function $F(L)$. It can be extracted by fitting a function to the points on the scatter plot. The function F typically contains constants used to set the firing rate to realistic values. These give us the freedom to normalize $D(\tau)$ in some convenient way, correcting for the arbitrary normalization by adjusting the parameters within F.

Static nonlinearities are used to introduce both firing thresholds and saturation into estimates of neural responses. Thresholds can be described by

10.5 Chapter Summary

We have presented a systematic treatment of exact and approximate maximum likelihood density estimation as a way of fitting probabilistic generative models and thereby performing representational learning. Recognition models, which are the statistical inverses of generative models, specify the causes underlying an input and play a crucial role in learning. We discussed the expectation maximization (EM) algorithm applied to invertible and noninvertible models, including the use of deterministic and probabilistic approximate recognition models and a lower bound on the log likelihood.

We presented a variety of models for continuous inputs with discrete, continuous, or vector-valued causes. These include mixture of Gaussians, K-means, factor analysis, principal components analysis, sparse coding, and independent components analysis. We also described the Helmholtz machine and discussed general issues of multi-resolution representation and coding.

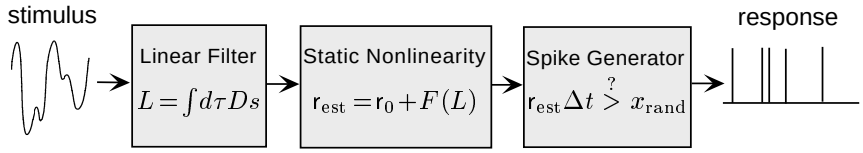

Figure 2.3 Simulating spiking responses to stimuli. The integral of the stimulus s times the optimal kernel D is first computed. The estimated firing rate is the background rate r_0 plus a nonlinear function of the output of the linear filter calculation. Finally, the estimated firing rate is used to drive a Poisson process that generates spikes.

not the estimate with the static nonlinearity $r_{est}(t) = r_0 + F(L(t))$. A theorem due to Bussgang (see appendix C) suggests that equation 2.6 will provide a reasonable kernel, even in the presence of a static nonlinearity, if the white noise stimulus used is Gaussian.

In some cases, the linear term of the Volterra series fails to predict the response even when static nonlinearities are included. Systematic improvements can be attempted by including more terms in the Volterra or Wiener series, but in practice it is quite difficult to go beyond the first few terms. The accuracy with which the first term, or first few terms, in a Volterra series can predict the responses of a neuron can sometimes be improved by replacing the parameter s in equation 2.7 with an appropriately chosen function of s, so that

$$L(t) = \int_0^\infty d\tau D(\tau) f(s(t-\tau)) \,. \tag{2.12}$$

A reasonable choice for this function is the response tuning curve. With this choice, the linear prediction is equal to the response tuning curve, $L = f(s)$, for static stimuli, provided that the integral of the kernel D is equal to 1. For time-dependent stimuli, we can think of equation 2.12 as a dynamic extension of the response tuning curve.

A model of the spike trains evoked by a stimulus can be constructed by using the firing-rate estimate of equation 2.8 to drive a Poisson spike generator (see chapter 1). Figure 2.3 shows the structure of such a model with a linear filter, a static nonlinearity, and a stochastic spike generator. In the figure, spikes are shown being generated by comparing the spiking probability $r(t)\,\Delta t$ to a random number, although the other methods discussed in chapter 1 could be used instead. Also, the linear filter acts directly on the stimulus s in figure 2.3, but it could act instead on some function $f(s)$, such as the response tuning curve.

2.3 Introduction to the Early Visual System

Before discussing how reverse-correlation methods are applied to visually responsive neurons, we review the basic anatomy and physiology of the

by coarsely quantizing the outputs of the highest spatial frequency filters generally has quite minimal perceptual consequences, while saving substantial coding cost (because these outputs are most numerous). This fact illustrates the important point that trying to build generative models of all aspects of visual images may be unnecessarily difficult, because only certain aspects of images are actually relevant. Unfortunately, abstract principles are unlikely to tell us what information in the input can safely be discarded independent of details of how the representations are to be used.

Overcomplete Representations

Sparse representations often have more output units than input units. Such representations, called overcomplete, are the subject of substantial work in multi-resolution theory. Many reasons have been suggested for overcompleteness, although none obviously emerges from the requirement of fitting good probabilistic models to input data.

One interesting idea comes from the notion that the task of manipulating representations should be invariant to the groups of symmetry transformations of the input, which, for images, include rotation, translation, and scaling. Complete representations are minimal, and so do not densely sample orientations. This means that the operations required to manipulate images of objects presented at angles not directly represented by the filters are different from those required at the represented angles (such as horizontal and vertical for the example of figure 10.10). When a representation is overcomplete in such a way that different orientations are represented roughly equally, as in primary visual cortex, the computational operations required to manipulate images are more uniform as a function of image orientation. Similar ideas apply across scale, so that the operations required to manipulate large and small images of the same object (as if viewed from near and far) are likewise similar. However, it is impossible to generate representations that satisfy all these constraints perfectly.

In more realistic models that include noise, other rationales for overcompleteness come from considering population codes, in which many units redundantly report information about closely related quantities so that uncertainty can be reduced. Despite the ubiquity of overcomplete population codes in the brain, there are few representational learning models that produce them satisfactorily. The coordinated representations required to construct population codes are often incompatible with other heuristics such as factorial or sparse coding.

Interdependent Causes

One of the failings of multi-resolution decompositions for coding is that the outputs are not mutually independent. This makes encoding each of

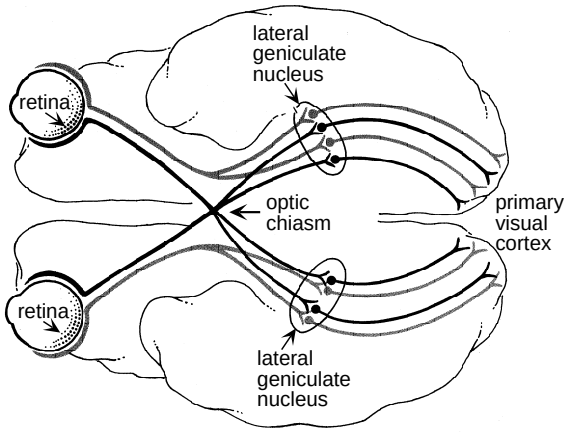

Figure 2.5 Pathway from the retina through the lateral geniculate nucleus (LGN) of the thalamus to the primary visual cortex in the human brain. (Adapted from Nicholls et al., 1992.)

of the two retinal ganglion cells shown are similar to those of the bipolar cells immediately above them in the figure, but now with superimposed action potentials. The two retinal ganglion cells shown in the figure have different responses and transmit different sequences of action potentials. G_2 fires while the light is on, and G_1 fires when it turns off. These are called ON and OFF responses, respectively. The optic nerve conducts the output spike trains of retinal ganglion cells to the lateral geniculate nucleus of the thalamus, which acts as a relay station between the retina and primary visual cortex (figure 2.5). Prior to arriving at the LGN, some retinal ganglion cell axons cross the midline at the optic chiasm. This allows the left and right sides of the visual fields from both eyes to be represented on the right and left sides of the brain, respectively (figure 2.5).

ON and OFF responses

Neurons in the retina, LGN, and primary visual cortex respond to light stimuli in restricted regions of the visual field called their receptive fields. Patterns of illumination outside the receptive field of a given neuron cannot generate a response directly, although they can significantly affect responses to stimuli within the receptive field. We do not consider such effects, although they are of considerable experimental and theoretical interest. In the monkey, cortical receptive fields range in size from around a tenth of a degree near the fovea to several degrees in the periphery. Within the receptive fields, there are regions where illumination higher than the background light intensity enhances firing, and other regions where lower illumination enhances firing. The spatial arrangement of these regions determines the selectivity of the neuron to different inputs. The term "receptive field" is often generalized to refer not only to the overall region where light affects neuronal firing, but also to the spatial and temporal structure within this region.

receptive fields

Visually responsive neurons in the retina, LGN, and primary visual cortex are divided into two classes, depending on whether or not the contribu-

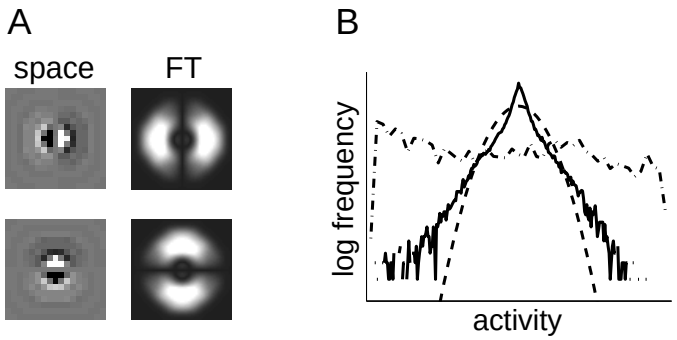

Figure 10.10 Multi-resolution filtering. (A) Vertical and horizontal filters (left) and their Fourier transforms (right) that are used at multiple positions and spatial scales to generate a multi-resolution representation. The rows of the matrix $\mathbf{W}$ are displayed here in gray scale on a two-dimensional grid representing the location of the corresponding input. (B) Log frequency distribution of the outputs of the highest spatial frequency filters (solid line) compared with a Gaussian distribution with the same mean and variance (dashed line) and the distribution of pixel values for the image shown in figure 10.11A (dot-dashed line). The pixel values of the image were rescaled to fit into the range. (Adapted from Simoncelli & Freeman, 1995; Karasaridis & Simoncelli, 1996.)

positions, we use them to consider various properties of representational learning from the perspective of information transmission and sparseness, overcompleteness, and residual dependencies between inferred causes.

Multi-resolution Decomposition

Many multi-resolution decompositions, with a variety of computational and representational properties, can be expressed as linear transformations $\mathbf{v} = \mathbf{W} \cdot \mathbf{u}$, where the rows of $\mathbf{W}$ describe filters, such as those illustrated in figure 10.10A. Figure 10.11 shows the result of applying multi-resolution filters, constructed by scaling and shifting the filters shown in figure 10.10A, to the photograph in figure 10.11A. Vertical and horizontal filters similar to those in figure 10.10A, but with different sizes, produce the decomposition shown in figures 10.11B-10.11D and 10.11F-10.11H when translated across the image. The level of gray indicates the output generated by placing the different filters over the corresponding point on the image. These outputs, plus the low-pass image in figure 10.11E and an extra high-pass image that is not shown, can be used to reconstruct the whole photograph almost perfectly through a generative process that is the inverse of the recognition process.

Coding

One reason for using multi-resolution decompositions is that they offer efficient ways of encoding visual images, whereas raw values of input pixels

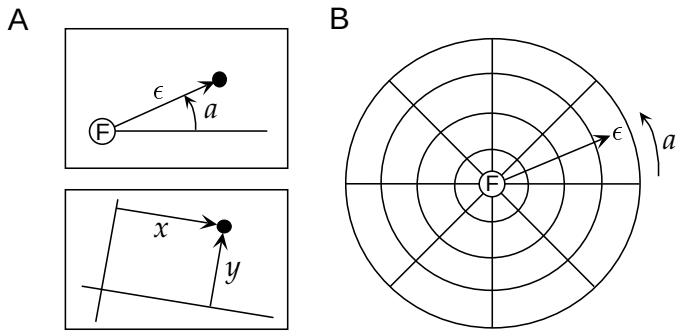

Figure 2.6 (A) Two coordinate systems used to parameterize image location. Each rectangle represents a tangent screen, and the filled circle is the location of a particular image point on the screen. The upper panel shows polar coordinates. The origin of the coordinate system is the fixation point F, the eccentricity ϵ is proportional to the radial distance from the fixation point to the image point, and a is the angle between the radial line from F to the image point and the horizontal axis. The lower panel shows Cartesian coordinates. The location of the origin for these coordinates and the orientation of the axes are arbitrary. They are usual chosen to center and align the coordinate system with respect to a particular receptive field being studied. (B) A bull's-eye pattern of radial lines of constant azimuth, and circles of constant eccentricity. The center of this pattern at zero eccentricity is the fixation point F. Such a pattern was used to generate the image in figure 2.7A.

Objects located a fixed distance from one eye lie on a sphere. Locations on this sphere can be represented using the same longitude and latitude angles used for the surface of the earth. Typically, the "north pole" for this spherical coordinate system is located at the fixation point, the image point that focuses onto the fovea or center of the retina. In this system of coordinates (figure 2.6), the latitude coordinate is called the eccentricity, ϵ, and the longitude coordinate, measured from the horizontal meridian, is called the azimuth, a. In primary visual cortex, the visual world is split in half, with the region $-90^\circ \leq a \leq +90^\circ$ for ϵ from 0° to about 70° (for both eyes) represented on the left side of the brain, and the reflection of this region about the vertical meridian represented on the right side of the brain.

eccentricity ϵ
azimuth a

In most experiments, images are displayed on a flat screen (called a tangent screen) that does not coincide exactly with the sphere discussed in the previous paragraph. However, if the screen is not too large, the difference is negligible, and the eccentricity and azimuth angles approximately coincide with polar coordinates on the screen (figure 2.6A). Ordinary Cartesian coordinates can also be used to identify points on the screen (figure 2.6A). The eccentricity ϵ and the x and y coordinates of the Cartesian system are based on measuring distances on the screen. However, it is customary to divide these measured distances by the distance from the eye to the screen and to multiply the result by $180^\circ/\pi$ so that these coordinates are ultimately expressed in units of degrees. This makes sense because it is the angular, not the absolute, size and location of an image that is typically relevant for studies of the visual system.

The EM algorithm for this noninvertible model would consist of alternately maximizing the function $\mathcal{F}$ given by

$$\mathcal{F}(\mathcal{W}, \mathcal{G}) = \left\langle \sum_{\mathbf{v}} Q[\mathbf{v}; \mathbf{u}, \mathcal{W}] \ln \frac{P[\mathbf{v}, \mathbf{u}; \mathcal{G}]}{Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]} \right\rangle \tag{10.44}$$

with respect to the parameters $\mathcal{W}$ and $\mathcal{G}$. For the M phase of the Helmholtz machine, this is exactly what is done. However, during the E phase, maximizing with respect to $\mathcal{W}$ is problematic because the function $Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]$ appears in two places in the expression for $\mathcal{F}$. This also makes the learning rule during the E phase take a different form from that during the M phase. Instead, the Helmholtz machine uses a simpler and more symmetric approximation to EM.

The approximation to EM used by the Helmholtz machine is constructed by re-expressing $\mathcal{F}$ from equation 10.10, explicitly writing out the average over input data and the expression for the Kullback-Leibler divergence,

$$\begin{aligned}\mathcal{F}(\mathcal{W}, \mathcal{G}) &= L(\mathcal{G}) - \sum_{\mathbf{u}} P[\mathbf{u}] D_{\mathrm{KL}}(Q[\mathbf{v}; \mathbf{u}, \mathcal{W}], P[\mathbf{v}|\mathbf{u}; \mathcal{G}]) \\ &= L(\mathcal{G}) - \sum_{\mathbf{u}} P[\mathbf{u}] \sum_{\mathbf{v}} Q[\mathbf{v}; \mathbf{u}, \mathcal{W}] \ln \left(\frac{Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]}{P[\mathbf{v}|\mathbf{u}; \mathcal{G}]} \right) . \end{aligned} \tag{10.45}$$

This is the function that is maximized with respect to $\mathcal{G}$ during the M phase for the Helmholtz machine. However, the E phase is not based on maximizing equation 10.45 with respect to $\mathcal{W}$. Instead, an approximate $\mathcal{F}$ function that we call $\tilde{\mathcal{F}}$ is used. This is constructed by using $P[\mathbf{u}; \mathcal{G}]$ as an approximation for $P[\mathbf{u}]$ and $D_{\mathrm{KL}}(P[\mathbf{v}|\mathbf{u}; \mathcal{G}], Q[\mathbf{v}; \mathbf{u}, \mathcal{W}])$ as an approximation for $D_{\mathrm{KL}}(Q[\mathbf{v}; \mathbf{u}, \mathcal{W}], P[\mathbf{v}|\mathbf{u}; \mathcal{G}])$ in equation 10.45. These are likely to be good approximations if the generative and approximate recognition models are accurate. Thus, we write

$$\begin{aligned}\tilde{\mathcal{F}}(\mathcal{W}, \mathcal{G}) &= L(\mathcal{G}) - \sum_{\mathbf{u}} P[\mathbf{u}; \mathcal{G}] D_{\mathrm{KL}}(P[\mathbf{v}|\mathbf{u}; \mathcal{G}], Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]) \\ &= L(\mathcal{G}) - \sum_{\mathbf{u}} P[\mathbf{u}; \mathcal{G}] \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}; \mathcal{G}] \ln \left(\frac{P[\mathbf{v}|\mathbf{u}; \mathcal{G}]}{Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]} \right) . \end{aligned} \tag{10.46}$$

and maximize this, rather than $\mathcal{F}$, with respect to $\mathcal{W}$ during the E phase. This amounts to averaging the "flipped" Kullback-Leibler divergence over samples of $\mathbf{u}$ created by the generative model, rather than real data samples. The advantage of making these approximations is that the E and M phases become highly symmetric, as can be seen by examining the second equalities in equations 10.45 and 10.46.

Learning in the Helmholtz machine proceeds by using stochastic sampling to replace the weighted sums in equations 10.45 and 10.46. In the M phase, an input $\mathbf{u}$ from $P[\mathbf{u}]$ is presented, and a sample $\mathbf{v}$ is drawn from the current recognition distribution $Q[\mathbf{v}; \mathbf{u}, \mathcal{W}]$. Then the generative weights $\mathcal{G}$ are changed according to the discrepancy between $\mathbf{u}$ and the generative or top-down prediction $\mathbf{f}(\mathbf{h} + \mathbf{G} \cdot \mathbf{v})$ of $\mathbf{u}$ (see the appendix). Thus, the generative model is trained to make $\mathbf{u}$ more likely to be generated by the cause

meridional ($\Delta\epsilon = 0$) displacements have the same cortical magnification factor $M(\epsilon)$, which itself is only a function of eccentricity. Consider two nearby image points with coordinates ϵ, a and $\epsilon + \Delta\epsilon$, a, separated by an angular distance $\Delta\epsilon$. The distance separating the activity evoked by these two image points on the cortex is ΔX. By the definition of the cortical magnification factor, these two quantities satisfy $\Delta X = M(\epsilon)\Delta\epsilon$ or, taking the limit as ΔX and $\Delta\epsilon$ go to 0,

$$\frac{dX}{d\epsilon} = M(\epsilon)\,. \tag{2.13}$$

For the macaque monkey, results such as figure 2.7A suggest that

$$M(\epsilon) = \frac{\lambda}{\epsilon_0 + \epsilon}\,, \tag{2.14}$$

with $\lambda \approx 12$ mm and $\epsilon_0 \approx 1^\circ$. Integrating equation 2.13 and defining $X = 0$ to be the point representing $\epsilon = 0$, we find

$$X = \lambda \ln(1 + \epsilon/\epsilon_0)\,. \tag{2.15}$$

For purely meridional displacements, the angular distance between two points at eccentricity ϵ with an azimuthal angle difference Δa is $\Delta a \epsilon \pi/180^\circ$. Here, the factor of ϵ corrects for the increase of arc length as a function of eccentricity, and the factor of $\pi/180^\circ$ converts ϵ from degrees to radians. The separation on the cortex, ΔY, corresponding to these points has a magnitude given by the cortical amplification times this distance. Taking the limit $\Delta a \to 0$, we find that

$$\frac{dY}{da} = -\frac{\epsilon\pi}{180^\circ} M(\epsilon)\,. \tag{2.16}$$

The minus sign in this relationship appears because the visual field is inverted on the cortex. Solving equation 2.16 gives

$$Y = -\frac{\lambda \epsilon a \pi}{(\epsilon_0 + \epsilon)180^\circ}\,. \tag{2.17}$$

The map defined by equations 2.15 and 2.17 is only approximate, particularly for small eccentricities. It is also not isotropic, which means that the magnification factor $M(\epsilon)$ only describes displacements for which either $\Delta\epsilon = 0$ or $\Delta a = 0$. Nevertheless, figure 2.7B shows that these coordinates agree fairly well with the map in figure 2.7A.

For eccentricities appreciably greater than 1°, equations 2.15 and 2.17 reduce to $X \approx \lambda \ln(\epsilon/\epsilon_0)$ and $Y \approx -\lambda\pi a/180^\circ$. These two formulas can be combined by defining the complex numbers $Z = X + iY$ and $z = (\epsilon/\epsilon_0)\exp(-i\pi a/180^\circ)$ (with i equal to the square root of -1) and writing $Z = \lambda \ln(z)$. For this reason, the cortical map is sometimes called a complex logarithmic map. For an image scaled radially by a factor γ, eccentricities change according to $\epsilon \to \gamma\epsilon$ while a is unaffected. Scaling of the eccentricity produces a shift $X \to X + \lambda \ln(\gamma)$ over the range of values where the simple logarithmic form of the map is valid. The logarithmic transformation thus causes images that are scaled radially outward on the retina to be represented at locations on the cortex translated in the X direction.

complex logarithmic map

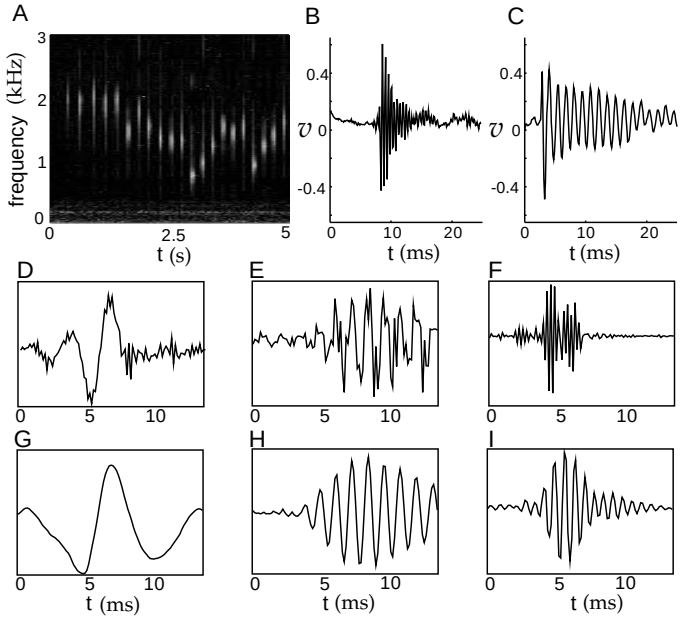

Figure 10.8 Independent components of tooth-tapping sounds. (A) Spectrogram of the input. (B-C) Waveforms for high- and low-frequency notes. The mouth acts as a damped resonant cavity in the generation of these tones. (D-F) Three independent components calculated on the basis of $1/80$ s samples taken from the input at random times. The graphs show the receptive fields (from $\mathbf{W}$) for three output units. D is reported to be sensitive to the sound of an air conditioner. E and F extract tooth taps of different frequencies. (G-I) The associated projective fields (from $\mathbf{G}$), showing the input activity associated with the causes in D-F. (Adapted from Bell & Sejnowski, 1996.)

tune by Beethoven. The input, sampled at 8 kHz, has the spectrogram shown in figure 10.8A. In this example, we have some idea about likely causes. For example, the plots in figures 10.8B and 10.8C show high- and low-frequency tooth taps, although other causes arise from the imperfect recording conditions (e.g., the background sound of an air conditioner). A close variant of the independent components analysis method described above was used to extract $N_v = 100$ independent components. Figure 10.8D, 10.8E, and 10.8F show the receptive fields of three of these components. The last two extract particular frequencies in the input. Figure 10.8G, 10.8H, and 10.8I show projective fields. Note that the projective fields are much smoother than the receptive fields.

Bell and Sejnowski (1997) also used visual input data similar to those used in the example of figure 10.7, along with the prior $p[v] \propto 1/\cosh(v)$, and found that independent components analysis extracts Gabor-like receptive fields similar to the projective fields shown in figure 10.7A.

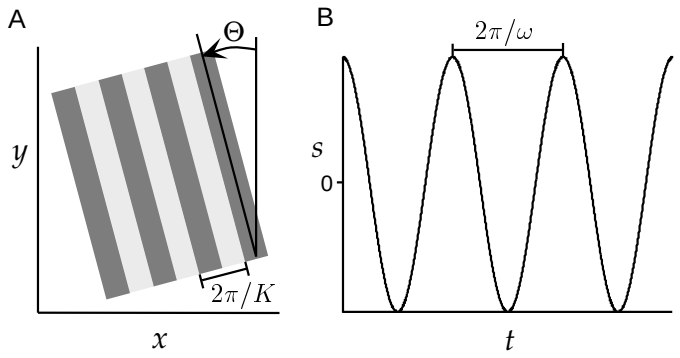

Figure 2.8 A counterphase grating. (A) A portion of a square-wave grating analogous to the sinusoidal grating of equation 2.18. The lighter stripes are regions where $s > 0$, and $s < 0$ within the darker stripes. K determines the wavelength of the grating and Θ, its orientation. Changing its spatial phase, Φ, shifts the entire light-dark pattern in the direction perpendicular to the stripes. (B) The light-dark intensity at any point of the spatial grating oscillates sinusoidally in time with period $2\pi/\omega$.

Of course, in practice a discrete approximation of such a stimulus must be used by dividing the image space into pixels and time into small bins. In addition, more structured random sets of images (randomly oriented bars, for example) are sometimes used to enhance the responses obtained during stimulation.

The Nyquist Frequency

Many factors limit the maximal spatial frequency that can be resolved by the visual system, but one interesting effect arises from the size and spacing of individual photoreceptors on the retina. The region of the retina with the highest resolution is the fovea at the center of the visual field. Within the macaque or human fovea, cone photoreceptors are densely packed in a regular array. Along any direction in the visual field, a regular array of tightly packed photoreceptors of size Δx samples points at locations $m\Delta x$ for $m = 1, 2, \ldots$. The (angular) frequency that defines the resolution of such an array is called the Nyquist frequency and is given by *Nyquist frequency*

$$K_{\text{nyq}} = \frac{\pi}{\Delta x}. \tag{2.20}$$

To understand the significance of the Nyquist frequency, consider sampling two cosine gratings with spatial frequencies of K and $2K_{\text{nyq}} - K$, with $K < K_{\text{nyq}}$. These are described by $s = \cos(Kx)$ and $s = \cos((2K_{\text{nyq}} - K)x)$. At the sampled points, these functions are identical because $\cos((2K_{\text{nyq}} - K)m\Delta x) = \cos(2\pi m - Km\Delta x) = \cos(-Km\Delta x) = \cos(Km\Delta x)$ by the periodicity and evenness of the cosine function (see figure 2.9). As a result, these two gratings cannot be distinguished by examining them only at the sampled points. Any two spatial frequencies $K < K_{\text{nyq}}$ and $2K_{\text{nyq}} - K$

Independent Components Analysis

As for the case of the mixtures of Gaussians model and factor analysis, an interesting model emerges from sparse coding as $\Sigma \to 0$. In this limit, the generative distribution (equation 10.30) approaches a δ function and always generates $\mathbf{u}(\mathbf{v}) = \mathbf{G} \cdot \mathbf{v}$. Under the additional restriction that there are as many causes as inputs, the approximation we used for the sparse coding model of making the recognition distribution deterministic becomes exact, and the recognition distribution that maximizes $\mathcal{F}$ is

$$Q[\mathbf{v}; \mathbf{u}] = |\det \mathbf{W}|^{-1} \delta(\mathbf{u} - \mathbf{W}^{-1} \cdot \mathbf{v}), \tag{10.35}$$

where $\mathbf{W} = \mathbf{G}^{-1}$ is the matrix inverse of the generative weight matrix. The factor $|\det \mathbf{W}|$ comes from the normalization condition on Q, $\int d\mathbf{v}\, Q(\mathbf{v}; \mathbf{u}) = 1$. At the maximum with respect to Q, the function $\mathcal{F}$ is

$$\mathcal{F}(Q, G) = \left\langle -\frac{1}{2\Sigma} |\mathbf{u} - \mathbf{G} \cdot \mathbf{W} \cdot \mathbf{u}|^2 + \sum_a g([\mathbf{W} \cdot \mathbf{u}]_a) \right\rangle + \ln |\det \mathbf{W}| + K, \tag{10.36}$$

where K is independent of $\mathbf{G}$. Under the conventional EM procedure, we would maximize this expression with respect to $\mathbf{G}$, keeping $\mathbf{W}$ fixed. However, the normal procedure fails in this case, because the minimum of the right side of equation 10.36 occurs at $\mathbf{G} = \mathbf{W}^{-1}$, and $\mathbf{W}$ is being held fixed, so $\mathbf{G}$ cannot change. This is an anomaly of coordinate ascent in this particular limit.

Fortunately, it is easy to fix this problem, because we know that $\mathbf{W} = \mathbf{G}^{-1}$ provides an exact inversion of the generative model. Therefore, instead of holding $\mathbf{W}$ fixed during the M phase of an EM procedure, we keep $\mathbf{W} = \mathbf{G}^{-1}$ at all times as we change $\mathbf{G}$. This sets $\mathcal{F}$ equal to the average log likelihood, and the process of optimizing with respect to $\mathbf{G}$ is equivalent to likelihood maximization. Because $\mathbf{W} = \mathbf{G}^{-1}$, maximizing with respect to $\mathbf{W}$ is equivalent to maximizing with respect to $\mathbf{G}$, and it turns out that this is easier to do. Therefore, we set $\mathbf{W} = \mathbf{G}^{-1}$ in equation 10.36, which causes the first term to vanish, and write the remaining terms as the log likelihood expressed as a function of $\mathbf{W}$ instead of $\mathbf{G}$,

$$L(\mathbf{W}) = \left\langle \sum_a g([\mathbf{W} \cdot \mathbf{u}]_a) \right\rangle + \ln |\det \mathbf{W}| + K. \tag{10.37}$$

Direct stochastic gradient ascent on this log likelihood can be performed using the update rule

$$\mathbf{W}_{ab} \to \mathbf{W}_{ab} + \epsilon \left([\mathbf{W}^{-1}]_{ba} + g'(v_a) u_b \right), \tag{10.38}$$

where ϵ is a small learning rate parameter, and we have used the fact that $\partial \ln |\det \mathbf{W}| / \partial \mathbf{W}_{ab} = [\mathbf{W}^{-1}]_{ba}$.

The update rule of equation 10.38 can be simplified by using a clever trick. Because $\mathbf{W}^T \mathbf{W}$ is a positive definite matrix (see the Mathematical

The spike-triggered average is related to the reverse-correlation function, as discussed in chapter 1, by

$$C(x, y, \tau) = \frac{Q_{rs}(x, y, -\tau)}{\langle r \rangle}, \tag{2.23}$$

where $\langle r \rangle$ is, as usual, the average firing rate over the entire trial, $\langle r \rangle = \langle n \rangle / T$.

To estimate the firing rate of a neuron in response to a particular image, we add a function of the output of a linear filter of the stimulus to the background firing rate r_0, as in equation 2.8, $r_{est}(t) = r_0 + F(L(t))$. As in equation 2.7, the linear estimate $L(t)$ is obtained by integrating over the past history of the stimulus with a kernel acting as the weighting function. Because visual stimuli depend on spatial location, we must decide how contributions from different image locations are to be combined to determine $L(t)$. The simplest assumption is that the contributions from different spatial points sum linearly, so that $L(t)$ is obtained by integrating over all x and y values:

linear response estimate

$$L(t) = \int_0^\infty d\tau \int dx dy \, D(x, y, \tau) s(x, y, t - \tau). \tag{2.24}$$

The kernel $D(x, y, \tau)$ determines how strongly, and with what sign, the visual stimulus at the point (x, y) and at time $t - \tau$ affects the firing rate of the neuron at time t. As in equation 2.6, the optimal kernel is given in terms of the firing rate-stimulus correlation function, or the spike-triggered average, for a white-noise stimulus with variance parameter σ_s^2 by

$$D(x, y, \tau) = \frac{Q_{rs}(x, y, -\tau)}{\sigma_s^2} = \frac{\langle r \rangle C(x, y, \tau)}{\sigma_s^2}. \tag{2.25}$$

The kernel $D(x, y, \tau)$ defines the space-time receptive field of a neuron. Because $D(x, y, \tau)$ is a function of three variables, it can be difficult to measure and visualize. For some neurons, the kernel can be written as a product of two functions, one that describes the spatial receptive field and the other, the temporal receptive field,

space-time receptive field

$$D(x, y, \tau) = D_s(x, y) D_t(\tau). \tag{2.26}$$

Such neurons are said to have separable space-time receptive fields. Separability requires that the spatial structure of the receptive field not change over time except by an overall multiplicative factor. When $D(x, y, \tau)$ cannot be written as the product of two terms, the neuron is said to have a nonseparable space-time receptive field. Given the freedom in equation 2.8 to set the scale of D (by suitably adjusting the function F), we typically normalize D_s so that its integral is 1, and use a similar rule for the components from which D_t is constructed. We begin our analysis by studying first the spatial and then the temporal components of a separable space-time receptive field, and then proceed to the nonseparable case. For simplicity, we ignore the possibility that cells can have slightly different receptive fields for the two eyes, which underlies the disparity tuning considered in chapter 1.

separable receptive field

nonseparable receptive field

projective field

impossible to construct receptive fields by averaging over these inputs in nonlinear models, such as sparse coding models. Furthermore, generative models are most naturally characterized by projective fields rather than receptive fields. The projective field associated with a particular cause v_a can be defined as the set of inputs that it frequently generates. This consists of all the $\mathbf{u}$ values for which $P[\mathbf{u}|v_a; \mathcal{G}]$ is sufficiently large when v_a is large. For the model of figure 10.1, the projective fields are simply the circles in figure 10.1C. It is important to remember that projective fields can be quite different from receptive fields.

Projective fields for the Olshausen and Field model trained on natural scenes are shown in figure 10.7A, with one picture for each component of $\mathbf{v}$. In this case, the projective field for v_a is simply the matrix elements G_{ab} plotted for all b values. In figure 10.7A, the index b is plotted over a two-dimensional grid representing the location of the input u_b within the visual field. The projective fields form a Gabor-like representation for images, covering a variety of spatial scales and orientations. The resemblance of this representation to the receptive fields of simple cells in primary visual cortex is quite striking, although these are the projective fields of the model, not its receptive fields. Unfortunately, there is no simple form for the receptive fields of the $\mathbf{v}$ units. Figure 10.7B compares the projective field of one unit with receptive fields determined by presenting either dots or gratings as inputs and recording the responses. The responses to the dots directly determine the receptive field, while responses to the gratings directly determine the Fourier transform of the receptive field. Differences between the receptive fields calculated on the basis of these two types of input are evident in the figure. In particular, the receptive field computed from gratings shows more spatial structure than the one mapped by dots. Nevertheless, both show a resemblance to the projective field and to a typical simple-cell receptive field.

In a generative model, projective fields are associated with the causes underlying the visual images presented during training. The fact that the causes extracted by the sparse coding model resemble Gabor patches within the visual field is somewhat strange from this perspective. It is difficult to conceive of images as arising from such low-level causes, instead of causes couched in terms of objects within the images, for example. From the perspective of good representation, causes that are more like objects and less like Gabor patches would be more useful. To put this another way, although the prior distribution over causes biased them toward mutual independence, the causes produced by the recognition model in response to natural images are not actually independent. This is due to the structure in images arising from more complex objects than bars and gratings. It is unlikely that this high-order structure can be extracted by a model with only one set of causes. It is more natural to think of causes in a hierarchical manner, with causes at a higher level accounting for structure in the causes at a lower level. The multiple representations in areas along the visual pathway suggests such a hierarchical scheme, but the corresponding models are still in the rudimentary stages of development.

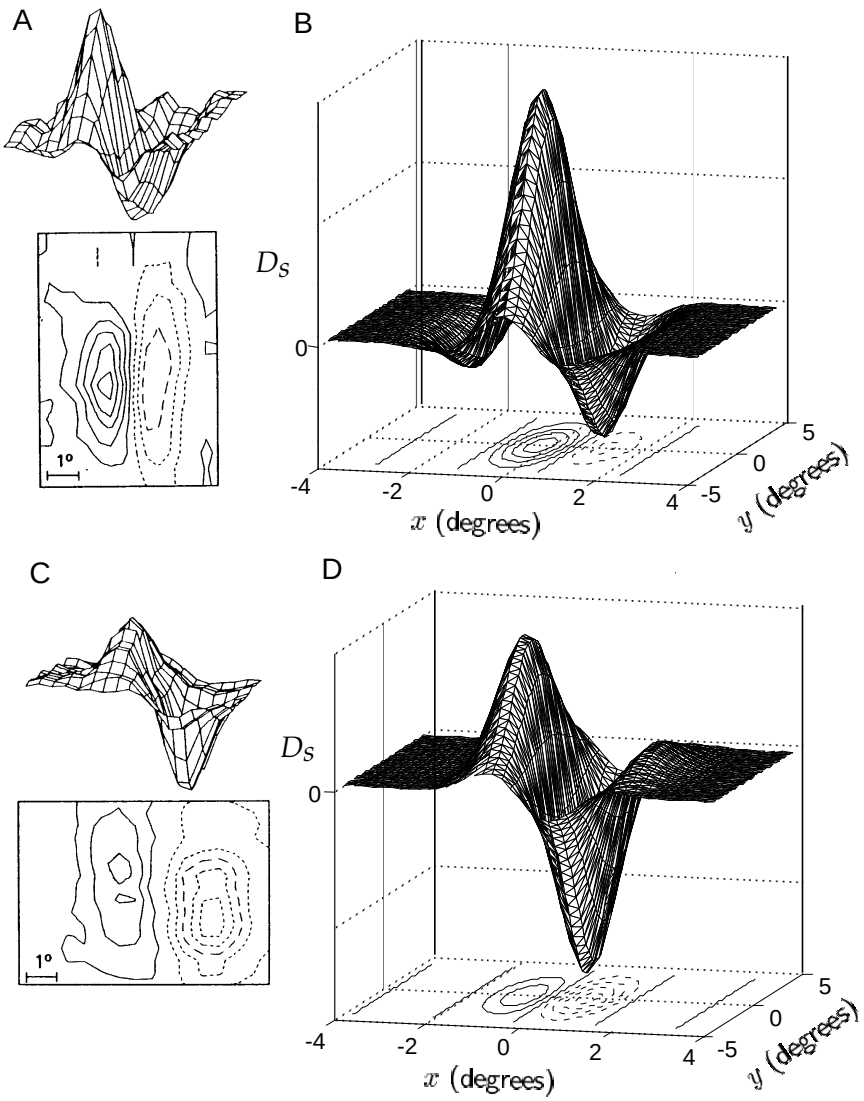

Figure 2.10 Spatial receptive field structure of simple cells. (A) and (C) Spatial structure of the receptive fields of two neurons in cat primary visual cortex determined by averaging stimuli between 50 ms and 100 ms prior to an action potential. The upper plots are three-dimensional representations, with the horizontal dimensions acting as the x-y plane and the vertical dimension indicating the magnitude and sign of $D_s(x, y)$. The lower contour plots represent the x-y plane. Regions with solid contour curves are ON areas where $D_s(x, y) > 0$, and regions with dashed contours are OFF areas where $D_s(x, y) < 0$. (B) and (D) Gabor functions (equation 2.27) with $\sigma_x = 1^\circ$, $\sigma_y = 2^\circ$, $1/k = 0.56^\circ$, and $\phi = 1 - \pi/2$ (B) or $\phi = 1 - \pi$ (D), chosen to match the receptive fields in A and C. (A and C adapted from Jones and Palmer, 1987a.)

perform factor analysis. An example of a function that provides a sparse prior is $g(v) = -\alpha|v|$. This generates a double exponential distribution (solid lines in figures 10.5B and 10.5C) similar to the activity distribution in figure 10.5A. Another commonly used form is

double exponential distribution

$$g(v) = -\ln(\beta^2 + v^2) \tag{10.31}$$

with β a constant, which generates a Cauchy distribution (dashed lines in figures 10.5B and 10.5C).

Cauchy distribution

For $g(v)$ such as equation 10.31, it is difficult to compute the recognition distribution $p[\mathbf{v}|\mathbf{u}; \mathcal{G}]$ exactly. This makes the sparse model noninvertible. Olshausen and Field chose a deterministic approximate recognition model. Thus, EM consists of finding $\mathbf{v}(\mathbf{u})$ during the E phase, and using it to adjust the parameters $\mathcal{G}$ during the M phase. To simplify the discussion, we make the covariance matrix of the generative model proportional to the identity matrix, $\mathbf{\Sigma} = \Sigma\mathbf{I}$. The function to be maximized is then

$$\mathcal{F}(\mathbf{v}(\mathbf{u}), \mathcal{G}) = \left\langle -\frac{1}{2\Sigma}|\mathbf{u} - \mathbf{G}\cdot\mathbf{v}(\mathbf{u})|^2 + \sum_{a=1}^{N_v} g(v_a(\mathbf{u})) \right\rangle + K, \tag{10.32}$$

where K is a term that is independent of $\mathbf{G}$ and $\mathbf{v}$. For convenience in discussing the EM procedure, we further take $\Sigma = 1$ and do not allow it to vary. Similarly, we assume that β in equation 10.31 is predetermined and held fixed. Then, $\mathcal{G}$ consists only of the matrix $\mathbf{G}$.

The E phase of EM involves maximizing $\mathcal{F}$ with respect to $\mathbf{v}(\mathbf{u})$ for every $\mathbf{u}$. This leads to the conditions (for all a)

$$\sum_{b=1}^{N_u}[\mathbf{u} - \mathbf{G}\cdot\mathbf{v}(\mathbf{u})]_b G_{ba} + g'(v_a) = 0\,. \tag{10.33}$$

The prime on $g(v_a)$ indicates a derivative. One way to solve this equation is to let $\mathbf{v}$ evolve over time according to the equation

$$\tau_{\mathbf{v}}\frac{dv_a}{dt} = \sum_{b=1}^{N_u}[\mathbf{u} - \mathbf{G}\cdot\mathbf{v}(\mathbf{u})]_b G_{ba} + g'(v_a)\,, \tag{10.34}$$

where $\tau_{\mathbf{v}}$ is an appropriate time constant. This equation changes $\mathbf{v}$ so that it asymptotically approaches a value $\mathbf{v} = \mathbf{v}(\mathbf{u})$ that satisfies equation 10.33 and sets the right side of equation 10.34 to 0. We assume that the evolution of $\mathbf{v}$ according to equation 10.34 is carried out long enough during the E phase for this to happen. This process is guaranteed to find only a local, not a global, maximum of $\mathcal{F}$, and it is not guaranteed to find the same local maximum on each iteration.

Equation 10.34 resembles the equation used in chapter 7 for a firing-rate network model. The term $\sum_b u_b G_{ba}$, which can be written in vector form as $\mathbf{G}^{\mathrm{T}}\cdot\mathbf{u}$, acts as the total input arising from units with activities $\mathbf{u}$ fed through a feedforward coupling matrix $\mathbf{G}^{\mathrm{T}}$. The term $-\sum_b[\mathbf{G}\cdot\mathbf{v}]_b G_{ba}$ can

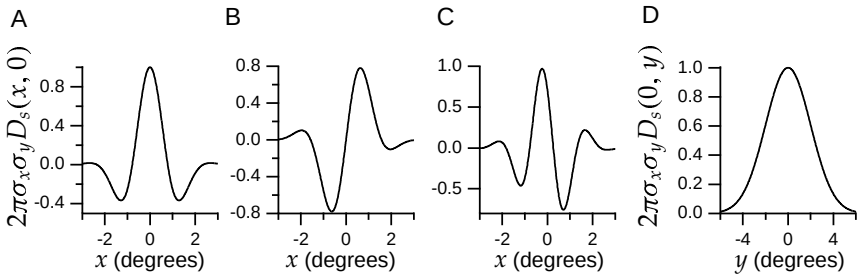

Figure 2.12 Gabor functions of the form given by equation 2.27. For convenience we plot the dimensionless function $2\pi\sigma_x\sigma_y D_s$. (A) A Gabor function with $\sigma_x = 1^\circ$, $1/k = 0.5^\circ$, and $\phi = 0$ plotted as a function of x for $y = 0$. This function is symmetric about $x = 0$. (B) A Gabor function with $\sigma_x = 1^\circ$, $1/k = 0.5^\circ$, and $\phi = \pi/2$ plotted as a function of x for $y = 0$. This function is antisymmetric about $x = 0$ and corresponds to using a sine instead of a cosine function in equation 2.27. (C) A Gabor function with $\sigma_x = 1^\circ$, $1/k = 0.33^\circ$, and $\phi = \pi/4$ plotted as a function of x for $y = 0$. This function has no particular symmetry properties with respect to $x = 0$. (D) The Gabor function of equation 2.27 with $\sigma_y = 2^\circ$ plotted as a function of y for $x = 0$. This function is simply a Gaussian.

making the substitutions $x \to x\cos(\theta) + y\sin(\theta)$ and $y \to y\cos(\theta) - x\sin(\theta)$ in equation 2.27. This produces a spatial receptive field that is maximally responsive to a grating with $\Theta = \theta$. Similarly, a receptive field centered at the point (x_0, y_0) rather than at the origin can be constructed by making the substitutions $x \to x - x_0$ and $y \to y - y_0$.

preferred orientation θ

rf center x_0, y_0

Temporal Receptive Fields

Figure 2.13 reveals the temporal development of the space-time receptive field of a neuron in the cat primary visual cortex through a series of snapshots of its spatial receptive field. More than 300 ms prior to a spike, there is little correlation between the visual stimulus and the upcoming spike. Around 210 ms before the spike ($\tau = 210$ ms), a two-lobed OFF-ON receptive field, similar to the ones in figure 2.10, is evident. As τ decreases (recall that τ measures time in a reversed sense), this structure first fades away and then reverses, so that the receptive field 75 ms before a spike has the opposite sign from what appeared at $\tau = 210$ ms. Due to latency effects, the spatial structure of the receptive field is less significant for $\tau < 75$ ms. The stimulus preferred by this cell is thus an appropriately aligned dark-light boundary that reverses to a light-dark boundary over time.

Reversal effects like those seen in figure 2.13 are a common feature of space-time receptive fields. Although the magnitudes and signs of the different spatial regions in figure 2.13 vary over time, their locations and shapes remain fairly constant. This indicates that the neuron has, to a good approximation, a separable space-time receptive field. When a space-time receptive field is separable, the reversal can be described by a function $D_t(\tau)$ that rises from 0, becomes positive, then negative, and ultimately

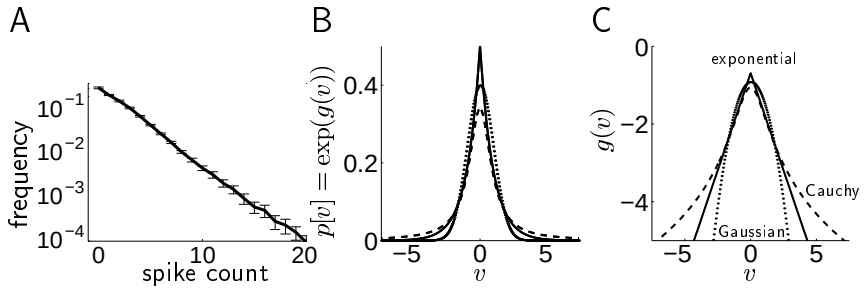

Figure 10.5 Sparse distributions. (A) Log frequency distribution for the activity of a macaque IT cell in response to video images. The fraction of times that various numbers of spikes appeared in a spike-counting window is plotted against the number of spikes. The size of the window was adjusted so that, on average, there were two spikes per window. (B) Three distributions $p[v] = \exp(g(v))$: double exponential ($g(v) = -|v|$, solid, kurtosis = 3); Cauchy ($g(v) = -\ln(1 + v^2)$, dashed, kurtosis = ∞); and Gaussian ($g(v) = -v^2/2$, dotted, kurtosis = 0). (C) The logarithms of the same three distributions. (A adapted from Baddeley et al., 1997.)

in chapter 4, so that the covariance matrix $\langle \mathbf{uu} \rangle$ is equal to the identity matrix, neither method will extract any structure at all from the input data. By contrast, the generative models discussed in the following sections produce non-Gaussian marginal distributions and attempt to account for structure in the input data beyond merely covariance (and the mean).

Sparse Coding

The v values in response to input in factor and principal components analysis tend to be Gaussian distributed. If we attempt to relate such causal variables to the activities of cortical neurons, we find a discrepancy, because the activity distributions of cortical cells in response to natural inputs are not Gaussian. Figure 10.5A shows an example of the distribution of the numbers of spikes counted within a particular time window for a neuron in the inferotemporal (IT) area of the macaque brain recorded while a monkey freely viewed television shows. The distribution is close to being exponential. This means that the neurons are most likely to fire a small number of spikes in the counting interval, but that they can occasionally fire a large number of spikes.

sparse distributions

Distributions that generate values for the components of $\mathbf{v}$ close to 0 most of the time, but occasionally far from 0, are called sparse. Intuitively, sparse distributions are more likely than Gaussians of the same mean and variance to generate values near 0, and also more likely to generate values far from 0. These occasional high values can convey substantial information. Distributions with this character are also called heavy-tailed. Figures 10.5B and 10.5C compare two sparse distributions with a Gaussian distribution.

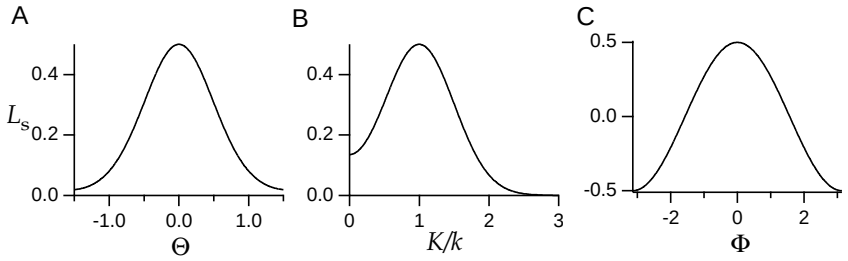

Figure 2.15 Selectivity of a Gabor filter with $\theta = \phi = 0$, $\sigma_x = \sigma_y = \sigma$, and $k\sigma = 2$ acting on a cosine grating with $A = 1$. (A) L_s as a function of stimulus orientation Θ for a grating with the preferred spatial frequency and phase, $K = k$ and $\Phi = 0$. (B) L_s as a function of the ratio of the stimulus spatial frequency to its preferred value, K/k, for a grating oriented in the preferred direction $\Theta = 0$ and with the preferred phase $\Phi = 0$. (C) L_s as a function of stimulus spatial phase Φ for a grating with the preferred spatial frequency and orientation, $K = k$ and $\Theta = 0$.

where

$$L_s = \int dx dy\, D_s(x, y) A \cos\left(Kx\cos(\Theta) + Ky\sin(\Theta) - \Phi\right) \tag{2.31}$$

and

$$L_t(t) = \int_0^\infty d\tau\, D_t(\tau) \cos\left(\omega(t - \tau)\right) . \tag{2.32}$$

The reader is invited to compute these integrals for the case $\sigma_x = \sigma_y = \sigma$. To show the selectivity of the resulting spatial receptive fields, we plot (in figure 2.15) L_s as functions of the parameters Θ, K, and Φ that determine the orientation, spatial frequency, and spatial phase of the stimulus. It is also instructive to write out L_s for various special parameter values. First, if the spatial phase of the stimulus and the preferred spatial phase of the receptive field are 0 ($\Phi = \phi = 0$), we find that

$$L_s = A \exp\left(-\frac{\sigma^2(k^2 + K^2)}{2}\right) \cosh\left(\sigma^2 kK\cos(\Theta)\right) , \tag{2.33}$$

which determines the orientation and spatial frequency tuning for an optimal spatial phase. Second, for a grating with the preferred orientation $\Theta = 0$ and a spatial frequency that is not too small, the full expression for L_s can be simplified by noting that $\exp(-\sigma^2 kK) \approx 0$ for the values of $k\sigma$ normally encountered (for example, if $K = k$ and $k\sigma = 2$, $\exp(-\sigma^2 kK) = 0.02$). Using this approximation, we find

$$L_s = \frac{A}{2} \exp\left(-\frac{\sigma^2(k - K)^2}{2}\right) \cos(\phi - \Phi) , \tag{2.34}$$

which reveals a Gaussian dependence on spatial frequency and a cosine dependence on spatial phase.

The amplitude of the sinusoidally oscillating linear response estimate (equation 2.32) is plotted as a function of the temporal frequency of the

phase produces a new value of $\mathbf{g}$ given by

$$\mathbf{g} = \frac{\langle v(\mathbf{u})\mathbf{u}\rangle}{\langle v^2(\mathbf{u})\rangle}\,. \tag{10.26}$$

This depends only on the covariance matrix of the input distribution, as does the more general form given in the appendix. Under EM, equations 10.24 and 10.26 are alternated until convergence.

For principal components analysis, we can say more about the value of $\mathbf{g}$ at convergence. We consider the case $|\mathbf{g}|^2 = 1$ because we can always multiply $\mathbf{g}$ and divide $v(\mathbf{u})$ by the same factor to make this true without affecting the dominant term in $\mathcal{F}(v(\mathbf{u}), \mathcal{G})$ as $\Sigma \to 0$. Then, the $\mathbf{g}$ that maximizes this dominant term must minimize

$$\left\langle |\mathbf{u} - \mathbf{g}(\mathbf{g}\cdot\mathbf{u})|^2 \right\rangle = \left\langle |\mathbf{u}|^2 - (\mathbf{g}\cdot\mathbf{u})^2 \right\rangle\,. \tag{10.27}$$

Here, we have used expression 10.24 for $v(\mathbf{u})$. Minimizing 10.27 with respect to $\mathbf{g}$, subject to the constraint $|\mathbf{g}|^2 = 1$, gives the result that $\mathbf{g}$ is the eigenvector of the covariance matrix $\langle \mathbf{u}\mathbf{u}\rangle$ with maximum eigenvalue. This is just the principal component vector and is equivalent to finding the vector of unit length with the largest possible average projection onto $\mathbf{u}$. Note that there are ways other than EM for finding eigenvectors of this matrix.

The argument we have given shows that principal components analysis is a degenerate form of factor analysis. This is also true if more than one factor is considered, although maximizing $\mathcal{F}$ constrains the projections $\mathbf{G}\cdot\mathbf{u}$ and therefore is sufficient only to force $\mathbf{G}$ to represent the principal components subspace of the data. The same subspace emerges from full factor analysis provided that the variances of all the factors are equal, even when they are nonzero.

Figure 10.4 illustrates an important difference between factor analysis and principal components analysis. In this example, $\mathbf{u}$ is a three-component input vector, $\mathbf{u} = (u_1, u_2, u_3)$. Just as in figure 10.3, samples of input data were generated on the basis of a "true" cause, v_{true} according to

$$u_b = v_{\text{true}} + \epsilon_b\,, \tag{10.28}$$

where ϵ_b represents the noise added to component b of the input. Input data points were generated from this equation by choosing a value of v_{true} from a Gaussian distribution with mean 0 and variance 1, and values of ϵ_b from independent Gaussian distributions with 0 means. The variances of the distributions for ϵ_1, ϵ_2, and ϵ_3 were all are equal to 0.25 in figures 10.4A and 10.4B. However, in figures 10.4C and 10.4D, the variance for ϵ_3 is much larger (equal to 9), as if the third sensor was much noisier than the other two sensors. The graphs in figure 10.4 show the mean of the values of v extracted from sample inputs by factor analysis, or the value of v for principal components analysis, as a function of the true value used to generate the data. Perfect extraction of the underlying cause would have $v = v_{\text{true}}$, but this is impossible in this case because of the noise. The

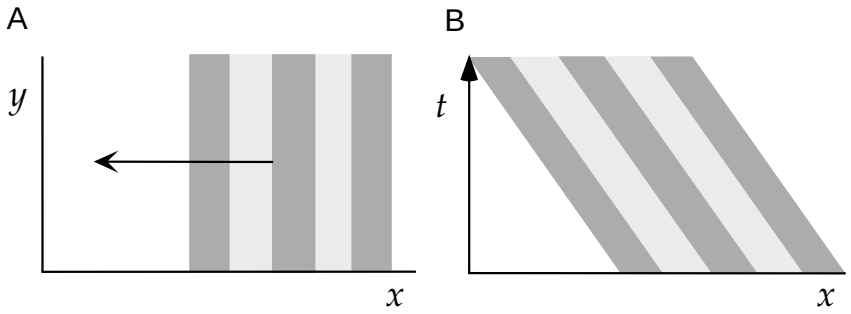

Figure 2.18 Space and space-time diagrams of a moving grating. (A) A vertically oriented grating moves to the left on a two-dimensional screen. (B) The space-time diagram of the image in A. The x location of the dark and light bands moves to the left as time progresses upward, representing the motion of the grating.

x-τ plot of a separable space-time kernel, similar to the one in figure 2.17A, generated by multiplying a Gabor function by the temporal kernel of equation 2.29.

We can also plot the visual stimulus in a space-time diagram, suppressing the y coordinate by assuming that the image does not vary as a function of y. For example, figure 2.18A shows a grating of vertically oriented stripes moving to the left on an x-y plot. In the x-t plot of figure 2.18B, this image appears as a series of sloped dark and light bands. These represent the projection of the image in figure 2.18A onto the x axis evolving as a function of time. The leftward slope of the bands corresponds to the leftward movement of the image.

Most neurons in primary visual cortex do not respond strongly to static images, but respond vigorously to flashed and moving bars and gratings. The receptive field structure of figure 2.17 reveals why this is the case, as is shown in figures 2.19 and 2.20. The image in figures 2.19A-C is a dark bar that is flashed on for a brief period of time. To describe the linear response estimate at different times, we consider a space-time receptive field similar to the one in figure 2.17A. The receptive field is positioned at three different times in figures 2.19A, B, and C. The height of the horizontal axis of the receptive field diagram indicates the time when the estimation is being made. Figure 2.19A corresponds to an estimate of $L(t)$ at the moment when the image first appears. At this time, $L(t) = 0$. As time progresses, the receptive field diagram moves upward. Figure 2.19B generates an estimate at the moment of maximum response, when the dark image overlaps the OFF area of the space-time receptive field, producing a positive contribution to $L(t)$. Figure 2.19C shows a later time when the dark image overlaps an ON region, generating a negative $L(t)$. The response for this flashed image is thus transient firing followed by suppression, as shown in Figure 2.19D.

Figures 2.19E and 2.19F show why a static dark bar is an ineffective stimulus. The static bar overlaps both the OFF region for small τ and the re-

discuss this limit for two clusters, as in figure 10.1. When Σ is extremely small, the recognition distribution $P[v|\mathbf{u}; \mathcal{G}]$ of equation 10.19 degenerates because it takes essentially two values, 0 or 1, depending on whether $\mathbf{u}$ is closer to one cluster or the other. This provides a deterministic, rather than a probabilistic, classification of $\mathbf{u}$. In the degenerate case, EM consists of choosing two random values for the centers of the two cluster distributions, and then repeatedly finding all the inputs $\mathbf{u}$ that are closest to a given center $\mathbf{g}_v$, and then moving $\mathbf{g}_v$ to the average of these points. This is called the K-means algorithm (with $K = 2$ for two clusters). The mixing proportions γ_v do not play an important role for the K-means algorithm. New input points are recognized as belonging to the clusters to which they are closest.

Factor Analysis

The model used in figure 10.3 is an example of factor analysis. In general, factor analysis involves a continuous vector of causes, $\mathbf{v}$, drawn from a Gaussian prior distribution, and uses a Gaussian generative distribution with a mean that depends linearly on $\mathbf{v}$. We assume that the distribution $p[\mathbf{u}]$ has a mean of 0 (nonzero means can be accommodated simply by shifting the input data). The defining distributions for factor analysis are thus

$$p[\mathbf{v}; \mathcal{G}] = \mathcal{N}(\mathbf{v}; \mathbf{0}, 1) \quad \text{and} \quad p[\mathbf{u}|\mathbf{v}; \mathcal{G}] = \mathcal{N}(\mathbf{u}; \mathbf{G} \cdot \mathbf{v}, \mathbf{\Sigma}), \tag{10.20}$$

where the extension of equation 10.18, expressed in terms of the mean $\mathbf{g}$ and covariance matrix $\mathbf{\Sigma}$, is

$$\mathcal{N}(\mathbf{u}; \mathbf{g}, \mathbf{\Sigma}) = \frac{1}{((2\pi)^{N_u} |\det \mathbf{\Sigma}|)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{u} - \mathbf{g}) \cdot \mathbf{\Sigma}^{-1} \cdot (\mathbf{u} - \mathbf{g})\right). \tag{10.21}$$

The expression $|\det \mathbf{\Sigma}|$ indicates the (absolute) value of the determinant of $\mathbf{\Sigma}$. In factor analysis, $\mathbf{\Sigma}$ is taken to be diagonal, $\mathbf{\Sigma} = \mathrm{diag}(\Sigma_1, \ldots, \Sigma_{N_u})$ (see the Mathematical Appendix), with all the diagonal elements strictly positive, so its inverse is simply $\mathbf{\Sigma}^{-1} = \mathrm{diag}(1/\Sigma_1, \ldots, 1/\Sigma_{N_u})$ and $|\det \mathbf{\Sigma}| = \Sigma_1 \Sigma_2 \ldots \Sigma_{N_u}$.

Because $\mathbf{\Sigma}$ is diagonal, the individual components of $\mathbf{v}$ are mutually independent. Thus, any correlations between the components of $\mathbf{u}$ must arise from the mean values $\mathbf{G} \cdot \mathbf{v}$ of the generative distribution. To be well specified, the model requires $\mathbf{v}$ to have fewer dimensions than $\mathbf{u}$ ($N_v < N_u$). In terms of heuristics, factor analysis seeks a relatively small number of independent causes that account, in a linear manner, for collective Gaussian structure in the inputs.

The recognition distribution for factor analysis has the Gaussian form

$$p[\mathbf{v}|\mathbf{u}; \mathcal{G}] = \mathcal{N}(\mathbf{v}; \mathbf{W} \cdot \mathbf{u}, \mathbf{\Psi}), \tag{10.22}$$

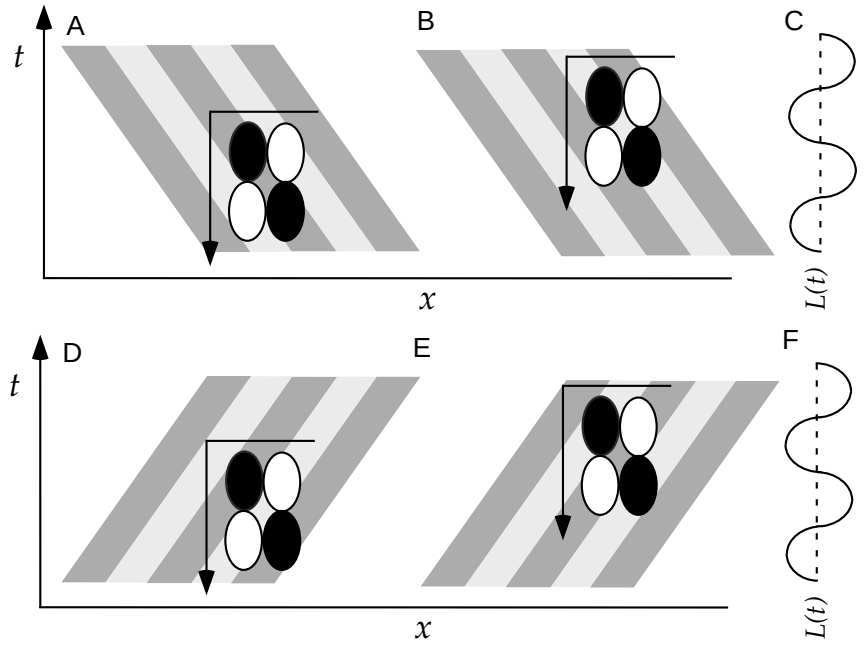

Figure 2.20 Responses to moving gratings estimated from a separable space-time receptive field. The receptive field is the same as in figure 2.19. (A-C) The stimulus is a grating moving to the left. At the time corresponding to A, OFF regions overlap with dark bands and ON regions with light bands, generating a strong response. At the time of the estimate in B, the alignment is reversed, and $L(t)$ is negative. (C) A plot of $L(t)$ versus time corresponding to the responses generated in A-B. Time runs vertically in this plot and $L(t)$ is plotted horizontally, with the dashed line indicating the zero axis and positive values plotted to the left. (D-F) The stimulus is a grating moving to the right. The responses are identical to those in A-C.

Nonseparable Receptive Fields

Many neurons in primary visual cortex are selective for the direction of motion of an image. Accounting for direction selectivity requires nonseparable space-time receptive fields. An example of a nonseparable receptive field is shown in figure 2.21A. This neuron has a three-lobed OFF-ON-OFF spatial receptive field, and these subregions shift to the left as time moves forward (and τ decreases). This means that the optimal stimulus for this neuron has light and dark areas that move toward the left. One way to describe a nonseparable receptive field structure is to use a separable function constructed from a product of a Gabor function for D_s and equation 2.29 for D_t, but to write these as functions of a mixture or rotation of the x and τ variables. The rotation of the space-time receptive field, as seen in figure 2.21B, is achieved by mixing the space and time coordinates, using the transformation

$$D(x, y, \tau) = D_s(x', y)D_t(\tau') \tag{2.35}$$

The M phase of EM consists, as always, of maximizing this expression with respect to $\mathcal{G}$. During the E phase we try to find the function $v(\mathbf{u})$ that maximizes $\mathcal{F}$. Because v is varied during the optimization procedure, the approach is sometimes called a variational method. The E and M steps make intuitive sense; we are finding the input-output relationship that maximizes the probability that the generative model would have simultaneously produced the input $\mathbf{u}$ and cause $v(\mathbf{u})$.

variational method

Noninvertible Probabilistic Models

The alternative to using a deterministic approximate recognition model is to treat $Q[v; \mathbf{u}]$ as a full probability distribution over v for each input example $\mathbf{u}$. In this case, we choose a specific functional form for Q, expressed in terms of a set of parameters collectively labeled $\mathcal{W}$. Thus, we write the approximate recognition distribution as $Q[v; \mathbf{u}, \mathcal{W}]$. Like generative models, approximate recognition models can have different structures and parameters. $\mathcal{F}$ can now be treated as a function of $\mathcal{W}$, rather than of Q, so we write it as $\mathcal{F}(\mathcal{W}, \mathcal{G})$. As in all cases, the M phase of EM consists of maximizing $F(\mathcal{W}, \mathcal{G})$ with respect to $\mathcal{G}$. The E phase now consists of maximizing $F(\mathcal{W}, \mathcal{G})$ with respect to $\mathcal{W}$. This has the effect of making $Q[v; \mathbf{u}, \mathcal{W}]$ as similar as possible to $P[v|\mathbf{u}; \mathcal{G}]$, in the sense that the KL divergence between them, averaged over the input data, is minimized (see equation 10.10).

In some cases, $\mathcal{W}$ has separate parameters for each possible input $\mathbf{u}$. This means that each input has a separate approximate recognition distribution which is individually tailored, subject to the inherent simplifying assumptions, to its own causes. The mean-field approximation to the Boltzmann machine discussed in chapter 7 is an example of this type.

It is not necessary to maximize $F(\mathcal{W}, \mathcal{G})$ completely with respect to $\mathcal{W}$ and then $\mathcal{G}$ during successive E and M phases. Instead, gradient ascent steps that modify $\mathcal{W}$ and $\mathcal{G}$ by small amounts can be taken in alternation, in which case the E and M phases effectively overlap.

Because each E and M step separately increases the value of $\mathcal{F}$, the EM algorithm is guaranteed to converge to at least a local maximum of $\mathcal{F}$, except in rare cases when the process of maximizing a function one coordinate at a time (which is called coordinate ascent) finds local maxima that other optimization methods avoid (we encounter an example of this later in the chapter). In general, the maximum found does not correspond to a local maximum of the likelihood function because Q is not exactly equal to the actual recognition distribution (that is, $\mathcal{F}$ is guaranteed only to be a lower bound on $L(\mathcal{G})$). Nevertheless, a good generative model should be obtained if the lower bound is tight.

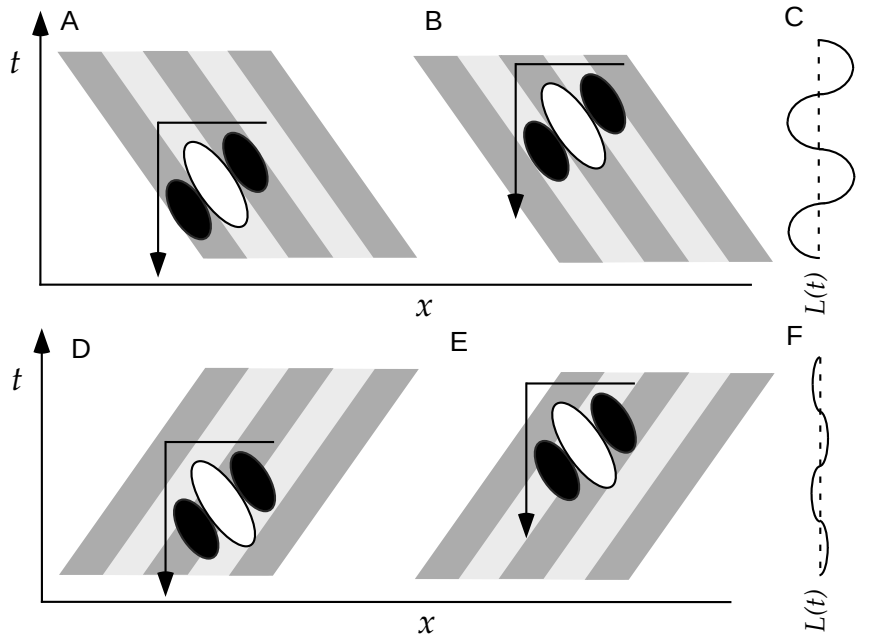

Figure 2.22 Responses to moving gratings estimated from a nonseparable space-time receptive field. Dark areas in the receptive field diagrams represent OFF regions and light areas, ON regions. (A-C) The stimulus is a grating moving to the left. At the time corresponding to A, OFF regions overlap with dark bands and the ON region overlaps a light band, generating a strong response. At the time of the estimate in B, the alignment is reversed, and $L(t)$ is negative. (C) A plot of $L(t)$ versus time corresponding to the responses generated in A and B. Time runs vertically in this plot, and $L(t)$ is plotted horizontally with the dashed line indicating the zero axis. (D-F) The stimulus is a grating moving to the right. Because of the tilt of the space-time receptive field, the alignment with the right-moving grating is never optimal and the response is weak (F).

Static Nonlinearities: Simple Cells

Once the linear response estimate $L(t)$ has been computed, the firing rate of a visually responsive neuron can be approximated by using equation 2.8, $\mathsf{r}_{\text{est}}(t) = \mathsf{r}_0 + F(L(t))$, where F is an appropriately chosen static nonlinearity. The simplest choice for F consistent with the positive nature of firing rates is rectification, $F = G[L]_+$, with G set to fit the magnitude of the measured firing rates. However, this choice makes the firing rate a linear function of the contrast amplitude, which does not match the data on the contrast dependence of visual responses. Neural responses saturate as the contrast of the image increases, and are more accurately described by $\mathsf{r} \propto A^n/(A_{1/2}^n + A^n)$ where n is near 2, and $A_{1/2}$ is a parameter equal to the contrast amplitude that produces a half-maximal response. This led Heeger (1992) to propose that an appropriate static nonlinearity to use is

contrast saturation

$$F(L) = \frac{G[L]_+^2}{A_{1/2}^2 + G[L]_+^2}, \tag{2.38}$$

definition of the Kullback-Leibler divergence to obtain

$$\begin{aligned}\mathcal{F}(Q, \mathcal{G}) &= \left\langle \sum_v Q[v; \mathbf{u}] \left(\ln p[\mathbf{u}; \mathcal{G}] + \ln \frac{P[v|\mathbf{u}; \mathcal{G}]}{Q[v; \mathbf{u}]} \right) \right\rangle \\ &= \langle \ln p[\mathbf{u}; \mathcal{G}] \rangle - \left\langle \sum_v Q[v; \mathbf{u}] \left(\ln \frac{Q[v; \mathbf{u}]}{P[v|\mathbf{u}; \mathcal{G}]} \right) \right\rangle \\ &= L(\mathcal{G}) - \left\langle D_{\mathrm{KL}}(Q[v; \mathbf{u}], P[v|\mathbf{u}; \mathcal{G}]) \right\rangle . \end{aligned} \tag{10.10}$$

Because the Kullback-Leibler divergence is never negative,

$$L(\mathcal{G}) \geq \mathcal{F}(Q, \mathcal{G}) , \tag{10.11}$$

and because $D_{\mathrm{KL}} = 0$ only if the two distributions being compared are identical, this inequality is saturated, becoming an equality, only if

$$Q[v; \mathbf{u}] = P[v|\mathbf{u}; \mathcal{G}] . \tag{10.12}$$

free energy $-\mathcal{F}$

The negative of $\mathcal{F}$ is related to the free energy used in statistical physics.

Expressions 10.10, 10.11, and 10.12 are critical to the operation of EM. The two phases of EM are concerned with separately maximizing (or at least increasing) $\mathcal{F}$ with respect to one of its two arguments, keeping the other one fixed. When $\mathcal{F}$ increases, this increases a lower bound on the log likelihood of the input data (equation 10.11). In the M phase, $\mathcal{F}$ is increased with respect to $\mathcal{G}$, keeping Q constant. For the generative models considered as examples in the previous section, it is possible to maximize $\mathcal{F}$ with respect to $\mathcal{G}$ in a single step. For other generative models, this may require multiple steps that perform gradient ascent on $\mathcal{F}$. In the E phase, $\mathcal{F}$ is increased with respect to Q, keeping $\mathcal{G}$ constant. From equation 10.10, we see that increasing $\mathcal{F}$ by changing Q is equivalent to reducing the average Kullback-Leibler divergence between $Q[v; \mathbf{u}]$ and $P[v|\mathbf{u}; \mathcal{G}]$. This makes $Q[v; \mathbf{u}]$ a better approximation of $P[v|\mathbf{u}; \mathcal{G}]$. The E phase can proceed in at least three possible ways, depending on the nature of the generative model being considered. We discuss these separately.

One advantage of EM over likelihood maximization through gradient methods is that large steps toward the maximum can be taken during each M cycle of modification. Of course, the log likelihood may have multiple maxima, in which case neither gradient ascent nor EM is guaranteed to find the globally optimal solution.

Invertible Models

If the causal model being considered is invertible, the E step of EM simply consists of solving equation 10.3 for the recognition distribution, and setting Q equal to the resulting $P[v|\mathbf{u}; \mathcal{G}]$, as in equation 10.12. This maximizes $\mathcal{F}$ with respect to Q by setting the Kullback-Leibler term in equation 10.10 to 0, and it makes the function $\mathcal{F}$ equal to $L(\mathcal{G})$, the average log

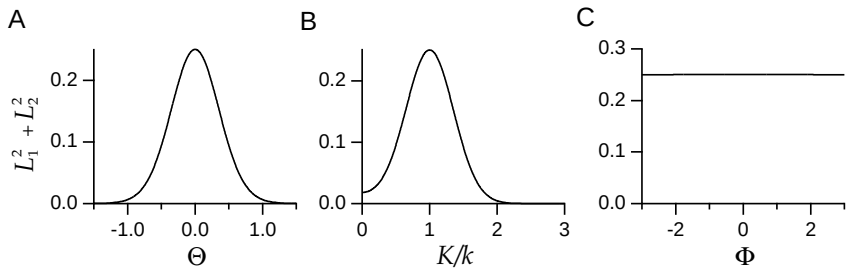

Figure 2.23 Selectivity of a complex cell model in response to a sinusoidal grating. The width and preferred spatial frequency of the Gabor functions underlying the estimated firing rate satisfy $k\sigma = 2$. (A) The complex cell response estimate, $L_1^2 + L_2^2$, as a function of stimulus orientation Θ for a grating with the preferred spatial frequency $K = k$. (B) $L_1^2 + L_2^2$ as a function of the ratio of the stimulus spatial frequency to its preferred value, K/k, for a grating oriented in the preferred direction $\Theta = 0$. (C) $L_1^2 + L_2^2$ as a function of stimulus spatial phase Φ for a grating with the preferred spatial frequency and orientation, $K = k$ and $\Theta = 0$.

where $B(\omega, K)$ is a temporal and spatial frequency-dependent amplitude factor. We do not need the explicit form of $B(\omega, K)$ here, but the reader is urged to derive it. For preferred spatial phase $\phi - \pi/2$,

$$L_2 = AB(\omega, K)\sin(\phi - \Phi)\cos(\omega t - \delta) \tag{2.40}$$

because $\cos(\phi - \pi/2 - \Phi) = \sin(\phi - \Phi)$. If we square and add these two terms, we obtain a result that does not depend on Φ,

$$L_1^2 + L_2^2 = A^2B^2(\omega, K)\cos^2(\omega t - \delta)\,, \tag{2.41}$$

because $\cos^2(\phi - \Phi) + \sin^2(\phi - \Phi) = 1$. Thus, we can describe the spatial-phase-invariant response of a complex cell by writing

$$\mathsf{r}(t) = \mathsf{r}_0 + G\left(L_1^2 + L_2^2\right)\,, \tag{2.42}$$

for some constant G. The selectivities of such a response estimate to grating orientation, spatial frequency, and spatial phase are shown in figure 2.23. The response of the model complex cell is tuned to orientation and spatial frequency, but the spatial phase dependence, illustrated for a simple cell in figure 2.15C, is absent. In computing the curve for figure 2.23C, we used the exact expressions for L_1 and L_2 from the integrals in equations 2.31 and 2.32, not the approximation used in equation 2.34 to simplify the previous discussion. Although it is not visible in the figure, there is a weak dependence on Φ when the exact expressions are used.

The complex cell response given by equations 2.41 and 2.42 reproduces the frequency-doubling effect seen in complex cell responses because the factor $\cos^2(\omega t - \delta)$ oscillates with frequency 2ω. This follows from the identity

$$\cos^2(\omega t - \delta) = \frac{1}{2}\cos\left(2(\omega t - \delta)\right) + \frac{1}{2}\,. \tag{2.43}$$

making $Q[v; \mathbf{u}]$ approximate $P[v|\mathbf{u}; \mathcal{G}]$ as accurately as possible, given the current parameters $\mathcal{G}$. We can include invertible models within the same general formalism used to describe noninvertible models by noting that, in the E phase for an invertible model, we simple set $Q[v; \mathbf{u}] = P[v|\mathbf{u}; \mathcal{G}]$ by solving equation 10.3.

Summary of Causal Models

In summary, causal models make use of the following probability distributions (for the case of continuous inputs and discrete causes).

- $p[\mathbf{u}]$, the input distribution
- $P[v; \mathcal{G}]$, the prior distribution over causes
- $p[\mathbf{u}|v; \mathcal{G}]$, the generative distribution
- $p[\mathbf{u}; \mathcal{G}]$, the marginal distribution
- $P[v|\mathbf{u}; \mathcal{G}]$, the recognition distribution
- $P[\mathbf{u}, v; \mathcal{G}]$, the joint distribution over inputs and causes
- $Q[v; \mathbf{u}]$, the approximate recognition distribution.

The goal of generative modeling, which is implemented by successive M phases of the EM algorithm, is to make $p[\mathbf{u}; \mathcal{G}] \approx p[\mathbf{u}]$ (as accurately as possible). This is done by using the marginal distribution obtained from prior E phases of EM and adjusting the parameters $\mathcal{G}$ to match it to the input distribution. The goal of each E phase is to make $Q[v; \mathbf{u}] \approx P[v|\mathbf{u}; \mathcal{G}]$ (as accurately as possible) for the current values of the generative parameters $\mathcal{G}$. Probabilistic recognition is carried out using the distribution $Q[v; \mathbf{u}]$ to determine the probability that cause v is responsible for input $\mathbf{u}$.

10.2 Density Estimation

density estimation

The process of matching the distribution $p[\mathbf{u}; \mathcal{G}]$ produced by the generative model to the actual input distribution $p[\mathbf{u}]$ is a form of density estimation. This technique is discussed in chapter 8 in connection with the Boltzmann machine. As mentioned in the introduction, the parameters $\mathcal{G}$ of the generative model are fitted to the input data by minimizing the discrepancy between the probability density of the input data $p[\mathbf{u}]$ and the marginal probability density $p[\mathbf{u}; \mathcal{G}]$ of equation 10.1. This discrepancy is measured using the Kullback-Leibler divergence (chapter 4),

$$\begin{aligned} D_{\mathrm{KL}}(p[\mathbf{u}], p[\mathbf{u}; \mathcal{G}]) &= \int d\mathbf{u}\, p[\mathbf{u}] \ln \frac{p[\mathbf{u}]}{p[\mathbf{u}; \mathcal{G}]} \\ &\approx -\langle \ln p[\mathbf{u}; \mathcal{G}] \rangle + K\,, \end{aligned} \tag{10.6}$$

2.6 Receptive Fields in the Retina and LGN

We end this discussion of the visual system by returning to the initial stages of the visual pathway and briefly describing the receptive field properties of neurons in the retina and LGN. Retinal ganglion cells display a wide variety of response characteristics, including nonlinear and direction-selective responses. However, a class of retinal ganglion cells (X cells in the cat or P cells in the monkey retina and LGN) can be described by a linear model built using reverse-correlation methods. The receptive fields of this class of retinal ganglion cells and an analogous type of LGN relay neurons are similar, so we do not treat them separately. The spatial structure of the receptive fields of these neurons has a center-surround structure consisting either of a circular central ON region surrounded by an annular OFF region, or the opposite arrangement of a central OFF region surrounded by an ON region. Such receptive fields are called ON-center and OFF-center, respectively. Figure 2.25A shows the spatial receptive fields of an ON-center cat LGN neuron.

The spatial structure of retinal ganglion and LGN receptive fields is well captured by a difference-of-Gaussians model in which the spatial receptive field is expressed as

difference of Gaussians

$$D_s(x, y) = \pm\left(\frac{1}{2\pi\sigma_{cen}^2}\exp\left(-\frac{x^2+y^2}{2\sigma_{cen}^2}\right) - \frac{B}{2\pi\sigma_{sur}^2}\exp\left(-\frac{x^2+y^2}{2\sigma_{sur}^2}\right)\right). \tag{2.45}$$

Here the center of the receptive field has been placed at $x = y = 0$. The first Gaussian function in equation 2.45 describes the center, and the second, the surround. The size of the central region is determined by the parameter σ_{cen}, while σ_{sur}, which is greater than σ_{cen}, determines the size of the surround. B controls the balance between center and surround contributions. The $\pm$ sign allows both ON-center $(+)$ and OFF-center $(-)$ cases to be represented. Figure 2.25B shows a spatial receptive field formed from the difference of two Gaussians that approximates the receptive field structure in figure 2.25A.

Figure 2.25C shows that the spatial structure of the receptive field reverses over time with, in this case, a central ON region reversing to an OFF region as τ increases. Similarly, the OFF surround region changes to an ON region with increasing τ, although the reversal and the onset are slower for the surround than for the central region. Because of the difference between the time course of the center and of the surround regions, the space-time receptive field is not separable, although the center and surround components are individually separable. The basic features of LGN neuron space-time receptive fields are captured by

$$D(x, y, \tau) = \pm\left(\frac{D_t^{cen}(\tau)}{2\pi\sigma_{cen}^2}\exp\left(-\frac{x^2+y^2}{2\sigma_{cen}^2}\right) - \frac{BD_t^{sur}(\tau)}{2\pi\sigma_{sur}^2}\exp\left(-\frac{x^2+y^2}{2\sigma_{sur}^2}\right)\right). \tag{2.46}$$

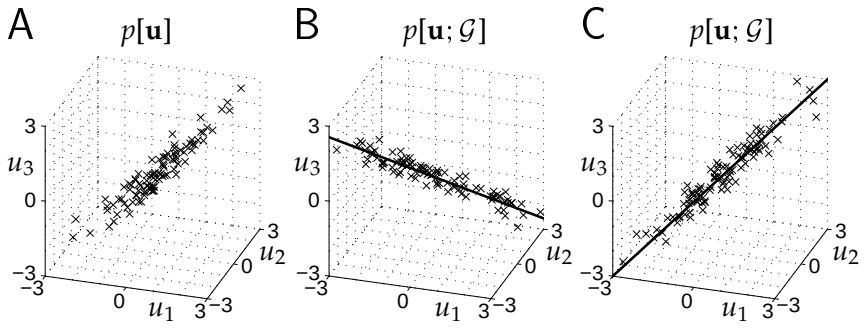

Figure 10.3 Factor analysis. (A) Input data points drawn from the distribution $p[\mathbf{u}]$ are indicated by the crosses. (B) The initial generative model. The solid line shows $\mathbf{g}$, and the crosses are synthetic data, which are samples from the generative distribution $p[\mathbf{u};\mathcal{G}]$ (with $\Sigma_a = 0.0625$, for all a). (C) The line $\mathbf{g}$ and synthetic data points generated by the optimal generative model.

shows intermediate results at three different times during the running of the EM procedure, starting from the generative model in figure 10.1B and ending up with the fit shown in figure 10.1C.

Continuous Generative Models

The data of figure 10.1A consist of two separated clusters of points, so a cause v that takes only two different values is appropriate. Figure 10.3A shows data that suggest the need for a continuous variable v. We can think of these data as the outputs of three noisy sensors, each measuring the same quantity. In this case, the cause v represents the value of the quantity being measured, and recognition corresponds to extracting this value from the sensor outputs. Because v is a continuous variable, the prior and recognition distributions in this case are probability densities, $p[v;\mathcal{G}]$ and $p[v|\mathbf{u};\mathcal{G}]$.

As for clustering, the generative model is determined by the prior distribution $p[v;\mathcal{G}]$ and the generative distribution $p[\mathbf{u}|v;\mathcal{G}]$, where $\mathbf{u} = (u_1, u_2, u_3)$ represents the three sensor readings. A simple choice for the prior distribution over v is a Gaussian with mean 0 and variance 1. The generative distribution is designed to capture the fact that the data points in figure 10.3A lie along a line in the three-dimensional space. It is the product of three Gaussian functions, one for each of the sensors, with means $g_a v$ and variances Σ_a for $a = 1, 2, 3$. The vector $\mathbf{g} = (g_1, g_2, g_3)$ specifies the direction of the line along which synthetic data points produced by the generative model lie, and the variances determine how tightly the points hug this line in each input dimension. In the example of figure 10.3A, the sensors all measure the same quantity. Thus, from an arbitrary initial $\mathbf{g}$, the generative model must find the best fit, $\mathbf{g} \propto (1, 1, 1)$.

Figure 10.3B shows synthetic data points generated from the generative

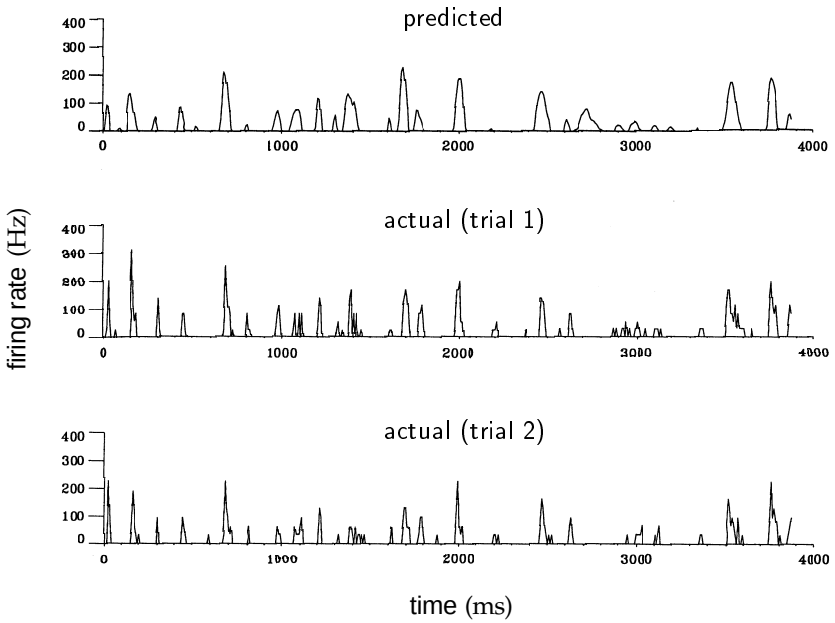

Figure 2.26 Comparison of predicted and measured firing rates for a cat LGN neuron responding to a video movie. The top panel is the rate predicted by integrating the product of the video image intensity and a linear filter obtained for this neuron from a spike-triggered average of a white-noise stimulus. The resulting linear prediction was rectified. The middle and lower panels are measured firing rates extracted from two different sets of trials. (Adapted from Dan et al., 1996.)

a rectifying static nonlinearity, was used to predict the firing rate of the neuron in response to a video movie. The top panel in figure 2.26 shows the resulting prediction, and the middle and lower panels show the actual firing rates extracted from two different groups of trials. The correlation coefficient between the predicted and actual firing rates was 0.5, which was very close to the correlation coefficient between firing rates extracted from different groups of trials. This means that the error of the prediction was no worse than the variability of the neural response itself.

2.7 Constructing V1 Receptive Fields

The models of visual receptive fields we have been discussing are purely descriptive, but they provide an important framework for studying how the circuits of the retina, LGN, and primary visual cortex generate neural responses. In an example of a more mechanistic model, Hubel and Wiesel (1962) proposed that the oriented receptive fields of cortical neurons could be generated by summing the input from appropriately selected LGN neurons. Their construction, shown in figure 2.27A, consists of alternating rows of ON-center and OFF-center LGN cells providing convergent input to a cortical simple cell. The left side of figure 2.27A shows the spatial ar-

Hubel-Wiesel simple cell model

distribution $p[\mathbf{u}|v; \mathcal{G}]$ that defines the generative model. Using Bayes theorem, it can be expressed in terms of the distributions that define the generative model as

$$P[v|\mathbf{u}; \mathcal{G}] = \frac{p[\mathbf{u}|v; \mathcal{G}]P[v; \mathcal{G}]}{p[\mathbf{u}; \mathcal{G}]}. \tag{10.3}$$

Once the recognition distribution has been computed from this equation, the probability of various causes being associated with a given input can be determined. For instance, in the example of figure 10.1, equation 10.3 can be used to determine that the point indicated by the filled square in figure 10.1C has probability $P[v = \mathrm{A}|\mathbf{u}; \mathcal{G}] = 0.8$ of being associated with neuron A and $P[v = \mathrm{B}|\mathbf{u}; \mathcal{G}] = 0.2$ of being associated with neuron B.

Recall that constructing a generative model involves making a number of assumptions about the nature of the causes underlying a set of inputs. The recognition model provides a mechanism for checking the self-consistency of these assumptions. This is done by examining the distribution of causes produced by the recognition model in response to actual data. This distribution should match the prior distribution over causes, and thus share its desired properties, such as mutual independence. If the prior distribution of the generative model does not match the actual distribution of causes produced by the recognition model, this is an indication that the imposed heuristic does not apply accurately to the input data.

Expectation Maximization

EM

During our discussion of generative models, we skipped over the process by which the parameters $\mathcal{G}$ are refined to optimize the match between synthetic and real input data. There are various ways of doing this. In this chapter (except for one case), we use an approach called expectation maximization (EM). The general theory of EM is discussed in detail in the following section but, as an introduction to the method, we apply it here to the example of figure 10.1. Recall that the problem of optimizing the generative model in this case involves adjusting the mixing proportions, means, and variances of the two Gaussian distributions until the clusters of synthetic data points in figure 10.1B and 10.1C match the clusters of actual data points in figure 10.1A.

To optimize the match between synthetic and real input data, the parameters $\mathbf{g}_v$ and $\mathbf{\Sigma}_v$, for v=A and B, of the Gaussian distributions of the generative model should equal the means and variances of the data points associated with each cluster in figure 10.1A. If we knew which cluster each input point belonged to, it would be a simple matter to compute these means and variances and construct the optimal generative model. Similarly, we could set γ_v, the prior probability of a given spike being a member of cluster v, equal to the fraction of data points assigned to that cluster. Of course, we do not know the cluster assignments of the input points; that would amount to knowing the answer to the recognition problem. However, we can make an informed guess about which point belongs to which

2.8 Chapter Summary

We continued from chapter 1 our study of the ways that neurons encode information, focusing on reverse-correlation analysis, particularly as applied to neurons in the retina, visual thalamus (LGN), and primary visual cortex. We used the tools of systems identification, especially the linear filter, Wiener kernel, and static nonlinearity, to build descriptive linear and nonlinear models of the transformation from dynamic stimuli to time-dependent firing rates. We discussed the complex logarithmic map governing the way that neighborhood relationships in the retina are transformed into cortex, Nyquist sampling in the retina, and Gabor functions as descriptive models of separable and nonseparable receptive fields. Models based on Gabor filters and static nonlinearities were shown to account for the basic response properties of simple and complex cells in primary visual cortex, including selectivity for orientation, spatial frequency and phase, velocity, and direction. Retinal ganglion cell and LGN responses were modeled using a difference-of-Gaussians kernel. We briefly described simple circuit models of simple and complex cells.

2.9 Appendices

A: The Optimal Kernel

Using equation 2.1 for the estimated firing rate, the expression in equation 2.3 to be minimized is

$$E = \frac{1}{T}\int_0^T dt \left(\mathsf{r}_0 + \int_0^\infty d\tau\, D(\tau)s(t-\tau) - \mathsf{r}(t)\right)^2 . \tag{2.48}$$

functional derivative

The minimum is obtained by setting the derivative of E with respect to the function D to 0. A quantity, such as E, that depends on a function, D in this case, is called a functional, and the derivative we need is a functional derivative. Finding the extrema of functionals is the subject of a branch of mathematics called the calculus of variations. A simple way to define a functional derivative is to introduce a small time interval Δt and evaluate all functions at integer multiples of Δt. We define $\mathsf{r}_i = \mathsf{r}(i\Delta t)$, $D_k = D(k\Delta t)$, and $s_{i-k} = s((i-k)\Delta t)$. If Δt is small enough, the integrals in equation 2.48 can be approximated by sums, and we can write

$$E = \frac{\Delta t}{T}\sum_{i=0}^{T/\Delta t}\left(\mathsf{r}_0 + \Delta t\sum_{k=0}^{\infty} D_k s_{i-k} - \mathsf{r}_i\right)^2 . \tag{2.49}$$

E is minimized by setting its derivative with respect to D_j for all values of j to 0,

$$\frac{\partial E}{\partial D_j} = 0 = \frac{2\Delta t}{T}\sum_{i=0}^{T/\Delta t}\left(\mathsf{r}_0 + \Delta t\sum_{k=0}^{\infty} D_k s_{i-k} - \mathsf{r}_i\right)s_{i-j}\,\Delta t\,. \tag{2.50}$$

sible causes (i.e., the number of clusters). Probabilistic methods can be used to make statistical inferences about the number of clusters in a data set, but they lie beyond the scope of this text.

Generative Models

To illustrate the concept of a generative model, we construct one (called a mixture of Gaussians model) for the data in figure 10.1A. The general form of the model is determined by the heuristics, assumptions about the nature of the causes and the way they generate inputs. However, the model has parameters that can be adjusted to fit the actual data that are observed. We begin by introducing parameters γ_A and γ_B that represent the proportions (also known as mixing proportions) of action potentials generated by each of the neurons. These might account for the fact that one of the neurons has a higher firing rate than the other, for example. The parameter γ_υ, with $\upsilon = \text{A}$ or B, specifies the probability $P[\upsilon; \mathcal{G}]$ that a given spike was generated by neuron υ in the absence of any knowledge about the input $\mathbf{u}$ associated with that spike. $P[\upsilon; \mathcal{G}] = \gamma_\upsilon$ is called the prior distribution over causes. The symbol $\mathcal{G}$ stands for all the parameters used to characterize the generative model. At this point, $\mathcal{G}$ consists of the two parameters γ_A and γ_B, but more parameters will be added as we proceed. We start by assigning γ_A and γ_B random values consistent with the constraint that they must sum to 1.

mixing proportions

prior $P[\upsilon; \mathcal{G}]$

parameters $\mathcal{G}$

To continue the construction of the generative model, we need to assume something about the distribution of $\mathbf{u}$ values arising from the action potentials generated by each neuron. An examination of figure 10.1A suggests that Gaussian distributions (with the same variance in both dimensions) might be appropriate. The probability density of $\mathbf{u}$ values given that neuron υ fired is $p[\mathbf{u}|\upsilon; \mathcal{G}]$. This is set to a Gaussian distribution with a mean and variance that, initially, we guess. The parameter list $\mathcal{G}$ now contains the prior probabilities, γ_A and γ_B, and the means and variances of the Gaussian distributions over $\mathbf{u}$ for $\upsilon = \text{A}$ and B, which we label $\mathbf{g}_\upsilon$ and Σ_υ, respectively. Note that Σ_υ is used to denote the variance of cluster υ, not its standard deviation, and also that each cluster is characterized by a single variance because we consider only circularly symmetric Gaussian distributions.

generative distribution $p[\mathbf{u}|\upsilon; \mathcal{G}]$

Figure 10.1B shows synthetic data points (crosses) generated by this model. To create each point, we set $\upsilon = \text{A}$ with probability $P[\upsilon = \text{A}; \mathcal{G}]$ (or otherwise set $\upsilon = \text{B}$) and then generated a point $\mathbf{u}$ randomly from the distribution $p[\mathbf{u}|\upsilon; \mathcal{G}]$. This generative model clearly has the capacity to create a data distribution with two clusters, similar to the one in figure 10.1A. However, the values of the parameters $\mathcal{G}$ used in figure 10.1B are obviously inappropriate. They must be adjusted by a learning procedure that matches, as accurately as possible, the distribution of synthetic data points in figure 10.1B to the actual input distribution in figure 10.1A. We describe how this is done in a later section. After optimization, as seen in

In terms of the Fourier transforms, equation 2.54 then becomes

$$\tilde{D}(\omega)\tilde{Q}_{ss}(\omega) = \tilde{Q}_{rs}(-\omega)\,, \tag{2.57}$$

which can be solved directly to obtain $\tilde{D}(\omega) = \tilde{Q}_{rs}(-\omega)/\tilde{Q}_{ss}(\omega)$. The inverse Fourier transform from which $D(\tau)$ is recovered is (Mathematical Appendix)

$$D(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega\, \tilde{D}(\omega)\exp(-i\omega\tau)\,, \tag{2.58}$$

so the optimal acausal kernel when the stimulus is temporally correlated is given by

$$D(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega\, \frac{\tilde{Q}_{rs}(-\omega)}{\tilde{Q}_{ss}(\omega)}\exp(-i\omega\tau)\,. \tag{2.59}$$

B: The Most Effective Stimulus

We seek the stimulus that produces the maximum predicted responses at time t subject to the fixed energy constraint

$$\int_0^T dt'\, \big(s(t')\big)^2 = \text{constant}\,. \tag{2.60}$$

We impose this constraint by the method of Lagrange multipliers (see the Mathematical Appendix), which means that we must find the unconstrained maximum value with respect to s of

$$\mathsf{r}_{\text{est}}(t) + \lambda\int_0^T dt'\, s^2(t') = \mathsf{r}_0 + \int_0^\infty d\tau\, D(\tau)s(t-\tau) + \lambda\int_0^T dt'\, \big(s(t')\big)^2\,, \tag{2.61}$$

where λ is the Lagrange multiplier. Setting the derivative of this expression with respect to the function s to 0 (using the same methods used in appendix A) gives

$$D(\tau) = -2\lambda s(t-\tau)\,. \tag{2.62}$$

The value of λ (which is less than 0) is determined by requiring that condition 2.60 is satisfied, but the precise value is not important for our purposes. The essential result is the proportionality between the optimal stimulus and $D(\tau)$.

C: Bussgang's Theorem

Bussgang (1952, 1975) proved that an estimate based on the optimal kernel for linear estimation can still be self-consistent (although not necessarily

recognition model

tification of the causes of particular images (e.g., object recognition) is performed by a second model, called the recognition model, that is constructed on the basis of the generative model. This procedure is analogous to analyzing an experiment by building a model of the processes thought to underly it, and using the model as a basis for extracting interesting features from the accumulated experimental data.

input vector $\mathbf{u}$
cause v
hidden or latent variable

We follow the convention of previous chapters and use the variables $\mathbf{u}$ and v to represent the input and output of the models we consider. The input vector $\mathbf{u}$ represents the data that we wish to analyze in terms of underlying causes. The output v, which is a variable that characterizes those causes, is sometimes called the hidden or latent variable, but we call it the "cause". In some models, causes may be associated with a vector $\mathbf{v}$, rather than a single number v.

recognition
deterministic recognition
probabilistic recognition

In terms of these variables, the ultimate goal of the models we consider is recognition, in which the model tells us something about the causes v underlying a particular input $\mathbf{u}$. Recognition can be either deterministic or probabilistic. In a model with deterministic recognition, the output $v(\mathbf{u})$ is the model's estimate of the cause underlying input $\mathbf{u}$. In probabilistic recognition, the model estimates the probability that different values of v are associated with input $\mathbf{u}$. In either case, the output is taken as the model's re-representation of the input.

heuristics

We consider models that infer causes in an unsupervised manner. This means that the existence and identity of any underlying causes must be deduced solely from two sources of information. One is the set of general assumptions and guesses, collectively known as heuristics, concerning the nature of the input data and the causes that might underly them. These heuristics determined the general form of the generative model. The other source of information is the statistical structure of the input data. In the absence of supervisory information or even reinforcement, causes are judged by their ability to explain and reproduce the statistical structure of the inputs they are designed to represent. The process is analogous to judging the validity of a model by comparing simulated data generated by it with the results of a real experiment. The basic idea is to use assumed causes to generate synthetic input data from a generative model. The statistical structure of the synthetic data is then compared with that of the real input data, and the parameters of the generative model are adjusted until the two are as similar as possible. If the final statistical match is good, the causes are judged trustworthy, and the model can be used as a basis for recognition.

Representational learning is a large and complex subject with a terminology and methodology that may be unfamiliar to many readers. Section 10.1 follows two illustrative examples to provide a general introduction. This should give the reader a basic idea of what representational learning attempts to achieve and how it works. Section 10.2 covers more technical aspects of the approach, and 10.3 surveys a number of examples.

(1996), and has been applied in an approach more closely related to the representational learning models of chapter 10 by Simoncelli & Schwartz (1999). The difference-of-Gaussians model for retinal and LGN receptive fields is due to Rodieck (1965) and Enroth-Cugell and Robson (1966). A useful reference on modeling of the early visual system is Wörgötter & Koch (1991). The issue of linearity and nonlinearity in early visual processing is reviewed by **Ferster (1994)**.

the link to dynamic programming, from Barto, Sutton & Anderson (1983), Watkins (1989), Barto,et al. (1990), **Bertsekas & Tsitsiklis (1996)**, and **Sutton & Barto (1998)**. **Bertsekas & Tsitsiklis (1996)** and **Sutton & Barto (1998)** describe some of the substantial theory of temporal difference learning that has been developed. Dynamic programming as a computational tool of ethology is elucidated in Mangel & Clark (1988).

Schultz (1998) reviews the data on the activity of primate dopamine cells during appetitive conditioning tasks, together with the psychological and pharmacological rationale for studying these cells. The link with temporal difference learning was made in Montague, Dayan & Sejnowski (1996), Friston et al. (1994), and Houk et al. (1995). **Houk et al. (1995)** reviews the basal ganglia from a variety of perspectives. Wickens (1993) provides a theoretically motivated treatment. The model in Montague et al. (1995) for Real's (1991) experiments in bumblebee foraging was based on Hammer's (1993) description of an octopaminergic neuron in honeybees that appears to play, for olfactory conditioning, a somewhat similar role to the primate dopaminergic cells.

The kernel representation of the weight between a stimulus and reward can be seen as a form of a serial compound stimulus (Kehoe, 1977) or a spectral timing model (Grossberg & Schmajuk, 1989). Grossberg and colleagues (see Grossberg, 1982, 1987, 1988) have developed a sophisticated mathematical model of conditioning, including aspects of opponent processing (Konorski, 1967; Solomon & Corbit, 1974), which puts prediction of the absence of reward (or the presence of punishment) on a more equal footing with prediction of the presence of reward, and develops aspects of how animals pay differing amounts of attention to stimuli. There are many other biologically inspired models of conditioning, particularly of the cerebellum (e.g., Gluck et al., 1990; **Gabriel & Moore, 1990**; Raymond et al., 1996; Mauk & Donegan, 1997).

3 Neural Decoding

3.1 Encoding and Decoding

In chapters 1 and 2, we considered the problem of predicting neural responses to known stimuli. The nervous system faces the reverse problem, determining what is going on in the real world from neuronal spiking patterns. It is interesting to attempt such computations ourselves, using the responses of one or more neurons to identify a particular stimulus or to extract the value of a stimulus parameter. We will assess the accuracy with which this can be done primarily by using optimal decoding techniques, regardless of whether the computations involved seem biologically plausible. Some biophysically realistic implementations are discussed in chapter 7. Optimal decoding allows us to determine limits on the accuracy and reliability of neuronal encoding. In addition, it is useful for estimating the information content of neuronal spike trains, an issue addressed in chapter 4.

As we discuss in chapter 1, neural responses, even to a single repeated stimulus, are typically described by stochastic models due to their inherent variability. In addition, stimuli themselves are often described stochastically. For example, the stimuli used in an experiment might be drawn randomly from a specified probability distribution. Natural stimuli can also be modeled stochastically as a way of capturing the statistical properties of complex environments.

Given this twofold stochastic model, encoding and decoding are related through a basic identity of probability theory called Bayes theorem. Let $\mathbf{r}$ represent the response of a neuron or a population of neurons to a stimulus characterized by a parameter s. Throughout this chapter, $\mathbf{r} = (r_1, r_2, \ldots, r_N)$ for N neurons is a list of spike-count firing rates, although, for the present discussion, it could be any other set of parameters describing the neuronal response. Several different probabilities and conditional probabilities enter into our discussion. A conditional probability is just an ordinary probability of an event occurring, except that its occurrence is subject to an additional condition. The conditional probability of event A occurring subject to the condition B is denoted by $P[A|B]$. The probabilities we need are:

conditional probability

tem, also the subsequent rewards. It would seem that the animal would have to consider optimizing whole sequences of actions, the number of which grows exponentially with time. Bellman's (1957) insight was that the Markov property effectively solves this problem. He rewrote equation 9.31 to separate the first and subsequent terms, and used a recursive principle for the latter. The Bellman equation is

$$v^*(u) = \max_a \left\{ \langle r_a(u) \rangle + \sum_{u'} P[u'|u;a] v^*(u') \right\} . \tag{9.32}$$

This says that maximizing reward at u requires choosing the action a that maximizes the sum of the mean immediate reward $\langle r_a(u) \rangle$ and the average of the largest possible values of all the states u' to which a can lead the system, weighted by their probabilities.

Policy Iteration

The actor-critic algorithm is a form of a dynamic programming technique called policy iteration. Policy iteration involves interleaved steps of policy evaluation (updating the critic) and policy improvement (updating the actor). Evaluation of policy $\mathbf{M}$ requires working out the values for all states u. We call these values $v^{\mathbf{M}}(u)$, to reflect explicitly their dependence on the policy. Each value is analogous to the quantity in 9.5. Using the same argument that led to the Bellman equation, we can derive the recursive formula

$$v^{\mathbf{M}}(u) = \sum_a P_{\mathbf{M}}[a;u] \left\{ \langle r_a(u) \rangle + \sum_{u'} P[u'|u;a] v^{\mathbf{M}}(u') \right\} . \tag{9.33}$$

Equation 9.33, for all states u, is a set of linear equations for $v^{\mathbf{M}}(u)$ that can be solved by matrix inversion. Reinforcement learning has been interpreted as a stochastic Monte Carlo method for performing this operation.

Monte Carlo method

Temporal difference learning uses an approximate Monte Carlo method to evaluate the right side of equation 9.33, and uses the difference between this approximation and the estimate of $v^{\mathbf{M}}(u)$ as the prediction error. The first idea underlying the method is that $r_a(u) + v^{\mathbf{M}}(u')$ is a sample whose mean is exactly the right side of equation 9.33. The second idea is bootstrapping, using the current estimate $v(u')$ in place of $v^{\mathbf{M}}(u')$ in this sample. Thus $r_a(u) + v(u')$ is used as a sampled approximation to $v^{\mathbf{M}}(u)$, and

$$\delta(t) = r_a(u) + v(u') - v(u) \tag{9.34}$$

is used as a sampled approximation to the discrepancy $v^{\mathbf{M}}(u) - v(u)$, which is an appropriate error measure for training $v(u)$ to equal $v^{\mathbf{M}}(u)$. Evaluating and improving policies from such samples without learning $P[u'|u;a]$ and $\langle r_a(u) \rangle$ directly is called an asynchronous, model-free approach to policy evaluation. It is possible to guarantee the convergence of

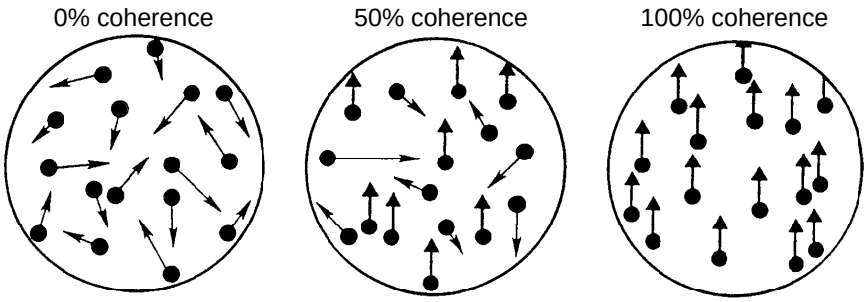

Figure 3.1 The moving random-dot stimulus for different levels of coherence. The visual image consists of randomly placed dots that jump every 45 ms according to the scheme described in the text. At 0% coherence the dots move randomly. At 50% coherence, half the dots move randomly and half move together (upward in this example). At 100% coherence all the dots move together. (Adapted from Britten et al., 1992.)

In the following sections, we present examples of decoding that involve both single neurons and neuronal populations. We first study a restricted case of single-cell decoding, discrimination between two different stimulus values. We then consider extracting the value of a parameter that characterizes a static stimulus from the responses of a population of neurons. As a final example, we return to single neurons and discuss spike-train decoding, in which an estimate of a time-varying stimulus is constructed from the spike train it evokes.

3.2 Discrimination

To introduce the notion of discriminability and the receiver operating characteristic that lie at the heart of discrimination analysis, we will discuss a fascinating study performed by Britten et al. (1992). In their experiments, a monkey was trained to discriminate between two directions of motion of a visual stimulus. The stimulus was a pattern of dots on a video monitor that jump from random initial locations to new locations every 45 ms. To introduce a sense of directed movement at a particular velocity, a percentage of the dots move together by a fixed amount in a fixed direction (figure 3.1). The coherently moving dots are selected randomly at each time step, and the remaining dots move to random new locations. The percentage of dots that move together in the fixed direction is called the coherence level. At 0% coherence, the image appears chaotic with no sense of any particular direction of motion. As the coherence increases, a sense of movement in a particular direction appears in the image until, at 100% coherence, the entire array of dots moves together on the monitor. By varying the degree of coherence, the task of detecting the movement direction can be made more or less difficult.

The experiments combined neural recording with behavioral measurements. In the behavioral part, the monkey had to report the direction

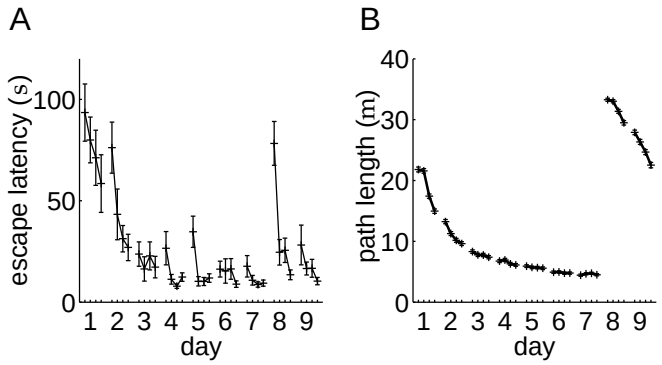

Figure 9.11 Comparison of rats and the model in the water maze task. (A) Average latencies of 12 rats in getting to a fixed platform in the water maze, using four trials per day. On the eighth day, the platform was moved to a new location, which is called reversal. (B) Average path length from 1000 simulations of the model performing the same task. Initial learning matches that of the rats, but performance is worse following reversal. (Adapted from Foster et al., 2000.)

model in a manner comparable to that of actual rats. Figure 9.11A shows the average performance of 12 real rats performing four trials per day in the water maze. The rats swam to a platform at a fixed location, starting from randomly chosen initial locations. The performance of the rats rapidly improves and levels off by about the sixth day. When the platform is moved on the eighth day, in what is called reversal training, the initial latency is long, because the rats search near the old platform position. However, they rapidly learn the new location. Figure 9.11B shows the performance of the model on the same task (though judged by path lengths rather than latencies). Initial learning is equally quick, with near perfect paths by the sixth day. However, performance during reversal training is poor, because the model has trouble forgetting the previous location of the platform. The rats are clearly better at handling this transition. Nevertheless the model shows something of the power of a primitive, but general, learning method.

9.5 Chapter Summary

We discussed reinforcement learning models for classical and instrumental conditioning, interpreting the former in terms of learning predictions about total future rewards and the latter in terms of optimization of those rewards. We introduced the Rescorla-Wagner or delta learning rule for classical conditioning, together with its temporal difference extension, and indirect and direct actor rules for instrumental conditioning given immediate rewards. Finally, we presented the actor-critic version of the dynamic programming technique of policy iteration, evaluating policies using temporal difference learning and improving them using the direct actor learning rule, based on surrogate immediate rewards from the evaluation step.

approximately Gaussian with the same variance, σ_r^2, but different means, $\langle r \rangle_+$ for the plus direction and $\langle r \rangle_-$ for the minus direction. A convenient measure of the separation between the distributions is the discriminability

discriminability d′

$$d' = \frac{\langle r \rangle_+ - \langle r \rangle_-}{\sigma_r}, \tag{3.4}$$

which is the distance between the means in units of their common standard deviation. The larger d', the more separated the distributions.

In the example we are considering, decoding involves using the neural response to determine in which of the two possible directions the stimulus moved. A simple decoding procedure is to determine the firing rate r during a trial and compare it to a threshold number z. If $r \geq z$, we report "plus"; otherwise we report "minus". Figure 3.2B suggests that if we choose z to lie somewhere between the two distributions, this procedure will give the correct answer at high coherence, but will have difficulty distinguishing the two directions at low coherence. This difficulty is clearly related to the degree to which the two distributions in figure 3.2B overlap, and thus to the discriminability.

The probability that the procedure outlined in the previous paragraph will generate the correct answer (called a hit) when the stimulus is moving in the plus direction is the conditional probability that $r \geq z$ given a plus stimulus, $P[r \geq z|+]$. The probability that it will give the answer "plus" when the stimulus is actually moving in the minus direction (called a false alarm) is similarly $P[r \geq z|-]$. These two probabilities completely determine the performance of the decoding procedure because the probabilities for the other two cases (reporting "minus" when the correct answer is "plus", and reporting "minus" when the correct answer is "minus") are $1-P[r \geq z|+]$ and $1-P[r \geq z|-]$, respectively. In signal detection theory, the quantity used to perform the discrimination, r in our case, is called the test, and the two probabilities corresponding to reporting a "plus" answer have specific names:

test size and power or false alarm and hit rate

$$\begin{array}{ll} \alpha(z) = P[r \geq z|-] & \text{is the size or false alarm rate of the test} \\ \beta(z) = P[r \geq z|+] & \text{is the power or hit rate of the test.} \end{array} \tag{3.5}$$

The following table shows how the probabilities of the test giving correct and incorrect answers in the different cases depend on α and β.

	probability	
stimulus	correct	incorrect
+	β	$1-\beta$
−	$1-\alpha$	α

The performance of the decoding procedure we have been discussing depends critically on the value of the threshold z to which the rate r is compared. Obviously, we would like to use a threshold for which the size

distant rewards are). Discounting has a major influence on the optimal behavior in problems for which there are many steps to a goal. Exponential discounting can be accommodated within the temporal difference framework by changing the prediction error δ to

$$\delta = r_a(u) + \gamma v(u') - v(u)\,, \tag{9.29}$$

which is then used in the learning rules of equations 9.26 and 9.28.

In computing the amount to change a weight or action value, we defined the worth of an action as the sum of the immediate reward delivered and the estimate of the future reward arising from the next state. A final generalization of actor-critic learning comes from basing the learning rules on the sum of the next two, three, or more immediate rewards delivered and basing the estimate of the future reward on more temporally distant times within a trial. As in discounting, we can use a factor λ to weight how strongly the expected future rewards from temporally distant points affect learning. Suppose that $\mathbf{u}(t) = \mathbf{u}(u(t))$ is the state vector used at time step t of a trial. Such generalized temporal difference learning can be achieved by computing new state vectors, defined by the recursive relation

$$\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}(t-1) + (1-\lambda)(\mathbf{u}(t) - \tilde{\mathbf{u}}(t-1))\,, \tag{9.30}$$

stimulus traces

TD(λ) rule

and using them instead of the original state vectors $\mathbf{u}$ in equations 9.26 and 9.28. These new state vectors $\tilde{\mathbf{u}}(t)$ are called stimulus traces, and the resulting learning rule is called the TD(λ) rule. Use of this rule with an appropriate value of λ can significantly speed up learning.

Learning the Water Maze

As an example of generalized reinforcement learning, we consider the water maze task. This is a navigation problem in which rats are placed in a large pool of milky water and have to swim around until they find a small platform that is submerged slightly below the surface of the water. The opaqueness of the water prevents them from seeing the platform directly, and their natural aversion to water (although they are competent swimmers) motivates them to find the platform. After several trials, the rats learn the location of the platform and swim directly to it when placed in the water.

Figure 9.10A shows the structure of the model, with the state vector $\mathbf{u}$ providing input to the critic and a collection of eight possible actions for the actor, which are expressed as compass directions. The components of $\mathbf{u}$ represent the activity of hippocampal place cells (which are discussed in chapters 1 and 8). Figure 9.10B shows the activation of one of the input units as a function of spatial position in the pool. The activity, like that of a place cell, is spatially restricted.

During training, each trial consists of starting the model rat from a random location at the outside of the maze and letting it run until it finds

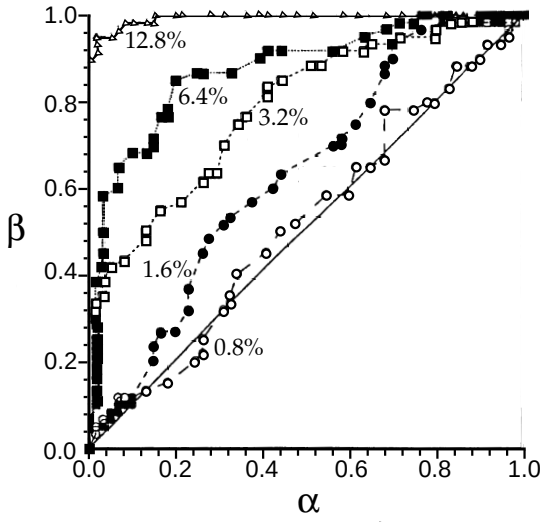

Figure 3.3 ROC curves for a variety of motion coherence levels. Each curve is the locus of points $(\alpha(z), \beta(z))$ for all z values. The values of α and β were computed from histograms such as those in figure 3.2B. The diagonal line is the ROC curve for random guessing. (Adapted from Britten et al., 1992.)

ROC curve is complicated by the fact that different threshold values can be used. This ambiguity can be removed by considering a slightly different task, called two-alternative forced choice. Here, the stimulus is presented twice, once with motion in the plus direction and once in the minus direction. The task is to decide which presentation corresponded to the plus direction, given the firing rates on both trials, r_1 and r_2. A natural extension of the test procedure we have been discussing is to answer trial 1 if $r_1 \geq r_2$ and otherwise answer trial 2. This removes the threshold variable from consideration.

two-alternative forced choice

In the two-alternative force-choice task, the value of r on one trial serves as the threshold for the other trial. For example, if the order of stimulus presentation is plus, then minus, the comparison procedure we have outlined will report the correct answer if $r_1 \geq z$ where $z = r_2$, and this has probability $P[r_1 \geq z|+] = \beta(z)$ with $z = r_2$. To determine the probability of getting the correct answer in a two-alternative forced-choice task, we need to integrate this probability over all possible values of r_2 weighted by their probability of occurrence. For small Δz, the probability that r_2 takes a value in the range between z and $z + \Delta z$ when the second trial has a minus stimulus is $p[z|-]\Delta z$, where $p[z|-]$ is the conditional firing-rate probability density for a firing rate $r = z$. Integrating over all values of z gives the probability of getting the correct answer,

$$P[\text{correct}] = \int_0^\infty dz\, p[z|-]\beta(z)\,. \tag{3.6}$$

Because the two-alternative forced-choice test is symmetric, this is also the

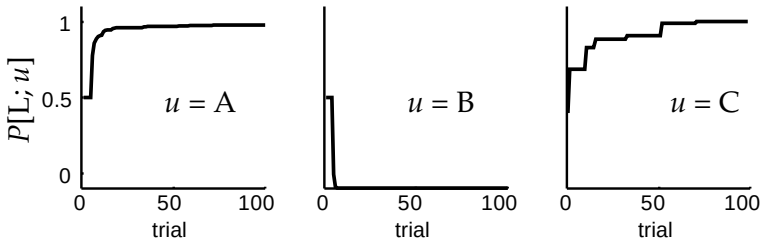

Figure 9.9 Actor-critic learning. The three curves show $P[\mathrm{L}; u]$ for the three starting locations u = A, B, and C in the maze of figure 9.7. These rapidly converge to their optimal values, representing left turns at A and C and a right turn at B. Here, $\epsilon = 0.5$ and $\beta = 1$.

is taken. This increases the chance that the rat makes the correct turn (left) at A in the maze of figure 9.7.

As the policy changes, the values, and therefore the temporal difference terms, change as well. However, because the values of all locations can increase only if we choose better actions at those locations, this form of policy improvement inevitably leads to higher values and better actions. This monotonic improvement (or at least not worsening) of the expected future rewards at all locations is proved formally in the dynamic programming theory of policy iteration for a class of problems called Markov decision problems (which includes the maze task), as discussed in the appendix.

Markov decision problems

Strictly speaking, policy evaluation should be complete before a policy is improved. It is also most straightforward to improve the policy completely before it is re-evaluated. A convenient (though not provably correct) alternative is to interleave partial policy evaluation and policy improvement steps. This is called the actor-critic algorithm. Figure 9.9 shows the result of applying this algorithm to the maze task. The plots show the development over trials of the probability of choosing to go left, $P[\mathrm{L}; u]$, for all three locations. The model rat quickly learns to go left at location A and right at B. Learning at location C is slow because the rat quickly learns that it is not worth going to C at all, so it rarely gets to try the actions there. Thus the algorithm makes an implicit choice of exploration strategy.

actor-critic algorithm

Generalizations of Actor-Critic Learning

The full actor-critic model for solving sequential action tasks includes three generalizations of the maze learner that we have presented. The first involves additional information that may be available at the different locations. If, for example, sensory information is available at a location u, we associate a state vector $\mathbf{u}(u)$ with that location. The vector $\mathbf{u}(u)$ parameterizes whatever information is available at location u that might help the animal decide which action to take. For example, the state vector might represent a scent of food that the rat might detect in the maze task. When a state vector is available, $v(u)$, which is the value at location

state vector $\mathbf{u}$

exactly opposite from the recorded neuron. In reality, the responses of this anti-neuron to a plus stimulus were just those of the recorded neuron to a minus stimulus, and vice versa. The idea of using the responses of a single neuron to opposite stimuli as if they were the simultaneous responses of two different neurons also reappears in our discussion of spike-train decoding. An observer predicting motion directions on the basis of just these two neurons at a level equal to the area under the ROC curve is termed an ideal observer.

Figure 3.2A shows a typical result for the performance of an ideal observer using one recorded neuron and its anti-neuron partner. The open circles in figure 3.2A were obtained by calculating the areas under the ROC curves for this neuron. Amazingly, the ability of the ideal observer to perform the discrimination task using a single neuron/anti-neuron pair is equal to the ability of the monkey to do the task. Although the choices of the ideal observer and the monkey do not necessarily match on a trial-to-trial basis, their performances are comparable when averaged over trials. This seems remarkable because the monkey presumably has access to a large population of neurons, while the ideal observer uses only two. One speculation is that correlations in the response variability between neurons limit the performance of the monkey.

The Likelihood Ratio Test

The discrimination test we have considered compares the firing rate to a threshold value. Could an observer do better than this already remarkable performance by comparing some other function of the firing rate to a threshold? What is the best test function to use for this purpose? The Neyman-Pearson lemma (proved in appendix A) shows that it is impossible to do better than to choose as the test function the ratio of probability densities (or, where appropriate, probabilities),

Neyman-Pearson lemma

$$l(r) = \frac{p[r|+]}{p[r|-]}, \tag{3.12}$$

which is known as the likelihood ratio. The test function r used above is not equal to the likelihood ratio. However, if the likelihood ratio is a monotonically increasing function of r, as it is for the data of Britten et al., the firing-rate threshold test is equivalent to using the likelihood ratio and is indeed optimal. Similarly, any monotonic function of the likelihood ratio will provide as good a test as the likelihood itself, and the logarithm is frequently used.

likelihood ratio

There is a direct relationship between the likelihood ratio and the ROC curve. As in equations 3.7 and 3.8, we can write

$$\beta(z) = \int_z^\infty dr\; p[r|+] \quad \text{so} \quad \frac{d\beta}{dz} = -p[z|+]\,. \tag{3.13}$$

at location u is given by $v(u) = w(u)$. This is an estimate of the total award that the rat expects to receive, on average, if it starts at the point u and follows its current policy through to the end of the maze. The average is taken over the stochastic choices of actions specified by the policy. In this case, the expected reward is simply equal to the weight. The learning procedure consists of two separate steps: policy evaluation, in which $w(u)$ is adjusted to improve the predictions of future reward, and policy improvement, in which $\mathbf{m}(u)$ is adjusted to increase the total reward.

Policy Evaluation

In policy evaluation, the rat keeps its policy fixed (i.e., keeps all the $\mathbf{m}(u)$ fixed) and uses temporal difference learning to determine the expected total future reward starting from each location. Suppose that, initially, the rat has no preference for turning left or right, that is, $\mathbf{m}(u) = 0$ for all u, so the probability of left and right turns is $1/2$ at all locations. By inspection of the possible places the rat can go, we find that the values of the states are

$$\begin{aligned} v(\mathrm{B}) &= \frac{1}{2}(0+5) = 2.5\,, \quad v(\mathrm{C}) = \frac{1}{2}(0+2) = 1\,, \quad \text{and} \\ v(\mathrm{A}) &= \frac{1}{2}(v(\mathrm{B}) + v(\mathrm{C})) = 1.75\,. \end{aligned} \tag{9.23}$$

These values are the average total future rewards that will be received during exploration of the maze when actions are chosen using the random policy. The temporal difference learning rule of equation 9.10 can be used to learn them. If the rat chooses action a at location u and ends up at location u', the temporal difference rule modifies the weight $w(u)$ by

critic learning rule

$$w(u) \to w(u) + \epsilon\delta \quad \text{with} \quad \delta = r_a(u) + v(u') - v(u)\,. \tag{9.24}$$

Figure 9.8 shows the result of applying the temporal difference rule to the maze task of figure 9.7. After a fairly short adjustment period, the weights $w(u)$ (and thus the predictions $v(u)$) fluctuate around the correct values for this policy, as given by equation 9.23. The size of the fluctuations could be reduced by making ϵ smaller, but at the expense of increasing the learning time.

In our earlier description of temporal difference learning, we included the possibility that the reward delivery might be stochastic. Here, that stochasticity is the result of a policy that makes use of the information provided by the critic. In the appendix, we discuss a Monte Carlo interpretation of the terms in the temporal difference learning rule that justifies its use.

We have thus far considered discriminating between two quite distinct stimulus values, plus and minus. Often we are interested in discriminating between two stimulus values $s+\Delta s$ and s that are very close to one another. In this case, the likelihood ratio is

$$\frac{p[r|s+\Delta s]}{p[r|s]} \approx \frac{p[r|s] + \Delta s \partial p[r|s]/\partial s}{p[r|s]}$$
$$= 1 + \Delta s \frac{\partial \ln p[r|s]}{\partial s} . \tag{3.18}$$

For small Δs, a test that compares

$$Z(r) = \frac{\partial \ln p[r|s]}{\partial s} \tag{3.19}$$

to a threshold $(z-1)/\Delta s$ is equivalent to the likelihood ratio test. The function $Z(r)$ is sometimes called the score. *score* $Z(r)$

3.3 Population Decoding

The use of large numbers of neurons to represent information is a basic operating principle of many nervous systems. Population coding has a number of advantages, including reduction of uncertainty due to neuronal variability and the ability to represent a number of different stimulus attributes simultaneously. Individual neurons in such a population typically have different but overlapping selectivities, so that many neurons, but not necessarily all, respond to a given stimulus. In the previous section, we discussed discrimination between stimuli on the basis of the response of a single neuron. The responses of a population of neurons can also be used for discrimination, with the only essential difference being that terms such as $p[r|s]$ are replaced by $p[\mathbf{r}|s]$, the conditional probability density of the population response $\mathbf{r}$. ROC analysis, likelihood ratio tests, and the Neyman-Pearson lemma continue to apply in exactly the same way. Discrimination is a special case of decoding in which only a few different stimulus values are considered. A more general problem is the extraction of a continuous stimulus parameter from one or more neuronal responses. In this section, we study how the value of a continuous parameter associated with a static stimulus can be decoded from the spike-count firing rates of a population of neurons.

Encoding and Decoding Direction

The cercal system of the cricket, which senses the direction of incoming air currents as a warning of approaching predators, is an interesting example of population coding involving a relatively small number of neurons. Crickets and related insects have two appendages called cerci extending

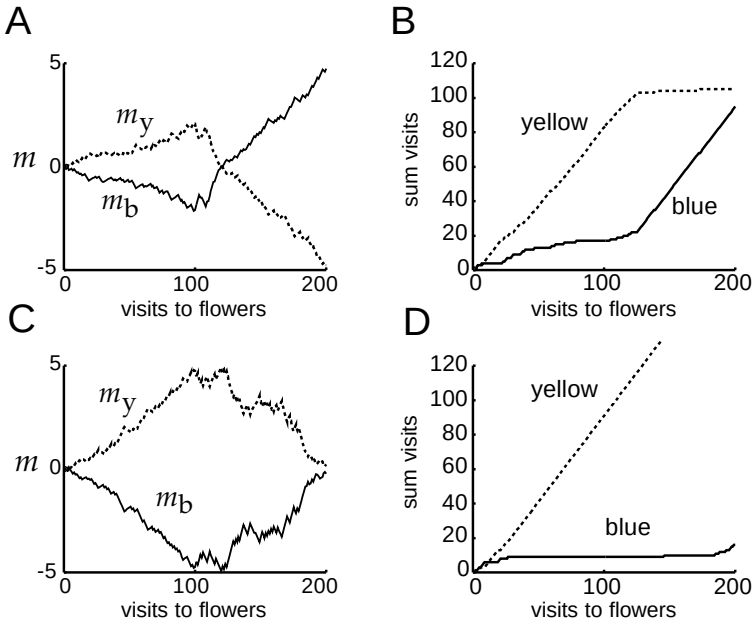

Figure 9.6 The direct actor. The statistics of the delivery of reward are the same as in figure 9.4, and $\epsilon = 0.1$, $\overline{r} = 1.5$, $\beta = 1$. The evolution of the weights and cumulative choices of flower type (with yellow dashed and blue solid) are shown for two sample sessions, one with good performance (A and B) and one with poor performance (C and D).

9.4 Sequential Action Choice

In the previous section, we considered ways that animals might learn to choose actions on the basis of immediate information about the consequences of those actions. A significant complication that arises when reward is based on a sequence of actions is illustrated by the maze task shown in figure 9.7. In this example, a hungry rat has to move through a maze, starting from point A, without retracing its steps. When it reaches one of the shaded boxes, it receives the associated number of food pellets and is removed from the maze. The rat then starts again at A. The task is to optimize the total reward, which in this case entails moving left at A and right at B. It is assumed that the animal starts knowing nothing about the structure of the maze or about the rewards.

If the rat started from point B or point C, it could learn to move right or left (respectively), using the methods of the previous section, because it experiences an immediate consequence of its actions in the delivery or nondelivery of food. The difficulty arises because neither action at the actual starting point, A, leads directly to a reward. For example, if the rat goes left at A and also goes left at B, it has to figure out that the former choice was good but the latter was bad. This is a typical problem in tasks that involve delayed rewards. The reward for going left at A is delayed

To determine the wind direction from the firing rates of the cercal interneurons, it is useful to change the notation somewhat. In place of the angle s, we can represent wind direction by a spatial vector $\vec{v}$ pointing parallel to the wind velocity and having unit length $|\vec{v}| = 1$ (we use over-arrows to denote spatial vectors). Similarly, we can represent the preferred wind direction for each interneuron by a vector $\vec{c}_a$ of unit length pointing in the direction specified by the angle s_a. In this case, we can use the vector dot product to write $\cos(s - s_a) = \vec{v} \cdot \vec{c}_a$. In terms of these vectors, the average firing rate is proportional to a half-wave rectified projection of the wind direction vector onto the preferred-direction axis of the neuron,

dot product

$$\left(\frac{f(s)}{r_{\max}}\right)_a = \left[\vec{v} \cdot \vec{c}_a\right]_+ . \tag{3.21}$$

Decoding the cercal system is particularly easy because of the close relationship between the representation of wind direction it provides and a two-dimensional Cartesian coordinate system. In a Cartesian system, vectors are parameterized by their projections onto x and y axes, v_x and v_y. These projections can be written as dot products of the vector being represented, $\vec{v}$, with vectors of unit length $\vec{x}$ and $\vec{y}$ lying along the x and y axes, $v_x = \vec{v} \cdot \vec{x}$ and $v_y = \vec{v} \cdot \vec{y}$. Except for the half-wave rectification, these equations are identical to equation 3.21. Furthermore, the preferred directions of the four interneurons, like the x and y axes of a Cartesian coordinate system, lie along two perpendicular directions (figure 3.5A). Four neurons are required, rather than two, because firing rates cannot represent negative projections. The cricket discovered the Cartesian coordinate system long before Descartes did, but failed to invent negative numbers! Perhaps credit should also be given to the leech, for Lewis and Kristan (1998) have shown that the direction of touch sensation in its body segments is encoded by four neurons in a virtually identical arrangement.

A vector $\vec{v}$ can be reconstructed from its Cartesian components through the component-weighted vector sum $\vec{v} = v_x\vec{x} + v_y\vec{y}$. Because the firing rates of the cercal interneurons we have been discussing are proportional to the Cartesian components of the wind direction vector, a similar sum should allow us to reconstruct the wind direction from a knowledge of the interneuron firing rates, except that four, not two, terms must be included. If r_a is the spike-count firing rate of neuron a, an estimate of the wind direction on any given trial can be obtained from the direction of the vector

$$\vec{v}_{\text{pop}} = \sum_{a=1}^{4} \left(\frac{r}{r_{\max}}\right)_a \vec{c}_a . \tag{3.22}$$

This vector is known as the population vector, and the associated decoding method is called the vector method. This decoding scheme works quite well. Figure 3.5B shows the root-mean-square difference between the direction determined by equation 3.22 and the actual wind direction that evoked the firing rates. The difference between the decoded and actual wind directions is around 6° except for dips at the angles corresponding to the preferred directions of the neurons. These dips are not due to the

population vector
vector method

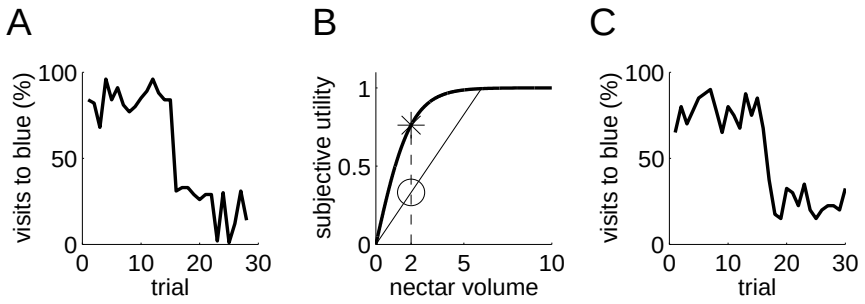

Figure 9.5 Foraging in bumblebees. (A) The mean preference of five real bumblebees for blue flowers over 30 trials involving 40 flower visits. There is a rapid switch of flower preference following the interchange of characteristics after trial 15. (B) Concave subjective utility function mapping nectar volume (in μl) to the subjective utility. The circle shows the average utility of the variable flowers, and the star shows the utility of the constant flowers. (C) The preference of a single model bee on the same task as the bumblebees. Here, $\epsilon = 3/10$ and $\beta = 23/8$. (Data in A from Real, 1991; B and C adapted from Montague et al., 1995.)

The Direct Actor

direct actor

An alternative to basing action choice on average rewards is to choose action values directly to maximize the average expected reward. The expected reward per trial is given in terms of the action probabilities and average rewards per flower by

$$\langle r \rangle = P[\mathrm{b}]\langle r_\mathrm{b} \rangle + P[\mathrm{y}]\langle r_\mathrm{y} \rangle \,. \tag{9.15}$$

This can be maximized by stochastic gradient ascent. To see how this is done, we take the derivative of $\langle r \rangle$ with respect to m_b,

$$\frac{\partial \langle r \rangle}{\partial m_\mathrm{b}} = \beta \left(P[\mathrm{b}]P[\mathrm{y}]\langle r_\mathrm{b} \rangle - P[\mathrm{y}]P[\mathrm{b}]\langle r_\mathrm{y} \rangle \right) . \tag{9.16}$$

In deriving this result, we have used the fact that, for the softmax distribution of equation 9.11,

$$\frac{\partial P[\mathrm{b}]}{\partial m_\mathrm{b}} = \beta P[\mathrm{b}]P[\mathrm{y}] \quad \text{and} \quad \frac{\partial P[\mathrm{y}]}{\partial m_\mathrm{b}} = -\beta P[\mathrm{y}]P[\mathrm{b}] \,. \tag{9.17}$$

Using the relation $P[\mathrm{y}] = 1 - P[\mathrm{b}]$, we can rewrite equation 9.16 in a form convenient for later use,

$$\frac{\partial \langle r \rangle}{\partial m_\mathrm{b}} = \beta P[\mathrm{b}](1 - P[\mathrm{b}])\langle r_\mathrm{b} \rangle - \beta P[\mathrm{y}]P[\mathrm{b}]\langle r_\mathrm{y} \rangle \,. \tag{9.18}$$

Furthermore, we can include an arbitrary parameter $\overline{r}$ in both these terms, because it cancels out. Thus,

$$\frac{\partial \langle r \rangle}{\partial m_\mathrm{b}} = \beta P[\mathrm{b}](1 - P[\mathrm{b}]) \left(\langle r_\mathrm{b} \rangle - \overline{r} \right) - \beta P[\mathrm{y}]P[\mathrm{b}] \left(\langle r_\mathrm{y} \rangle - \overline{r} \right) . \tag{9.19}$$

locity and acceleration. This complicates the interpretation of their activity as reporting movement direction in a particular coordinate system.

Unlike the cercal interneurons, M1 neurons do not have orthogonal preferred directions that form a Cartesian coordinate system. Instead, the preferred directions of the neurons appear to point in all directions with roughly equal probability. If the projection axes are not orthogonal, the Cartesian sum of equation 3.22 is not the correct way to reconstruct $\vec{v}$. Nevertheless, if the preferred directions point uniformly in all directions and the number of neurons N is sufficiently large, the population vector

$$\vec{v}_{\rm pop} = \sum_{a=1}^{N} \left(\frac{r - r_0}{r_{\rm max}} \right)_a \vec{c}_a \tag{3.24}$$

will, on average, point in a direction parallel to the arm movement direction vector $\vec{v}$. If we average equation 3.24 over trials and use equation 3.23, we find

$$\langle \vec{v}_{\rm pop} \rangle = \sum_{a=1}^{N} (\vec{v} \cdot \vec{c}_a) \vec{c}_a \, . \tag{3.25}$$

We leave as an exercise the proof that $\langle \vec{v}_{\rm pop} \rangle$ is approximately parallel to $\vec{v}$ if a large enough number of neurons is included in the sum, and if their preferred-direction vectors point randomly in all directions with equal probability. Later in this chapter, we discuss how corrections can be made if the distribution of preferred directions is not uniform or the number of neurons is not large. The population vectors constructed from equation 3.24 on the basis of responses of neurons in primary motor cortex, recorded while a monkey performed a reaching task, are compared with the actual directions of arm movements in figure 3.6.

Optimal Decoding Methods

The vector method is a simple decoding method that can perform quite well in certain cases, but it is neither a general nor an optimal way to reconstruct a stimulus from the firing rates of a population of neurons. In this section, we discuss two methods that can, by some measure, be considered optimal. These are called Bayesian inference and maximum a posteriori (MAP) inference. We also discuss a special case of MAP called maximum likelihood (ML) inference. The Bayesian approach involves finding the minimum of a loss function that expresses the cost of estimation errors. MAP inference and ML inference generally produce estimates that are as accurate, in terms of the variance of the estimate, as any that can be achieved by a wide class of estimation methods (so-called unbiased estimates), at least when large numbers of neurons are used in the decoding. Bayesian and MAP estimates use the conditional probability that a stimulus parameter takes a value between s and $s + \Delta s$, given that the set of N encoding neurons fired at rates given by $\mathbf{r}$. The probability density

We consider two simple methods of solving the bee foraging task. In the first method, called the indirect actor, the bee learns to estimate the expected nectar volumes provided by each flower by using a delta rule. It then bases its action choice on these estimates. In the second method, called the direct actor, the choice of actions is based directly on maximizing the expected average reward.

The Indirect Actor

indirect actor

One course for the bee to follow is to learn the average nectar volumes provided by each type of flower and base its action choice on these. This is called an indirect actor scheme, because the policy is mediated indirectly by the expected volumes. Here, this means setting the action values to

$$m_b = \langle r_b \rangle \quad \text{and} \quad m_y = \langle r_y \rangle \; . \tag{9.13}$$

In our discussion of classical conditioning, we saw that the Rescorla-Wagner or delta rule develops weights that approximate the average value of a reward, just as required for equation 9.13. Thus, if the bee chooses a blue flower on a trial and receives nectar volume r_b, it should update m_b according to the prediction error by

$$m_b \to m_b + \epsilon\delta \quad \text{with} \quad \delta = r_b - m_b \, , \tag{9.14}$$

and leave m_y unchanged. If it lands on a yellow flower, m_y is changed to $m_y + \epsilon\delta$ with $\delta = r_y - m_y$, and m_b is unchanged. If the probability densities of reward $p[r_b]$ and $p[r_y]$ change slowly relative to the learning rate, m_b and m_y will track $\langle r_b \rangle$ and $\langle r_y \rangle$, respectively.

Figure 9.4 shows the performance of the indirect actor on the two-flower foraging task. Figure 9.4A shows the course of weight change due to the delta rule in one example run. Figures 9.4B-D indicate the quality of the action choice by showing cumulative sums of the number of visits to blue and yellow flowers in three different runs. For ideal performance in this task, the dashed line would have slope 1 until trial 100 and 0 thereafter, and the solid line would show the reverse behavior, close to what is seen in figure 9.4C. This reflects the consistent choice of the optimal flower in both halves of the trial. A value of $\beta = 1$ (figure 9.4B) allows for continuous exploration, but at the cost of slow learning. When $\beta = 50$ (figure 9.4C & D), the tendency to exploit sometimes leads to good performance (figure 9.4C), but other times, the associated reluctance to explore causes the policy to perform poorly (figure 9.4D).

Figure 9.5A shows action choices of real bumblebees in a foraging experiment. This experiment was designed to test risk aversion in the bees, so the blue and yellow flowers differed in the reliability rather than the quantity of their nectar delivery. For the first 15 trials (each involving 40 visits to flowers), blue flowers always provided 2 μl of nectar, whereas $\frac{1}{3}$ of the yellow flowers provided 6 μl, and $\frac{2}{3}$ provided nothing (note that the mean

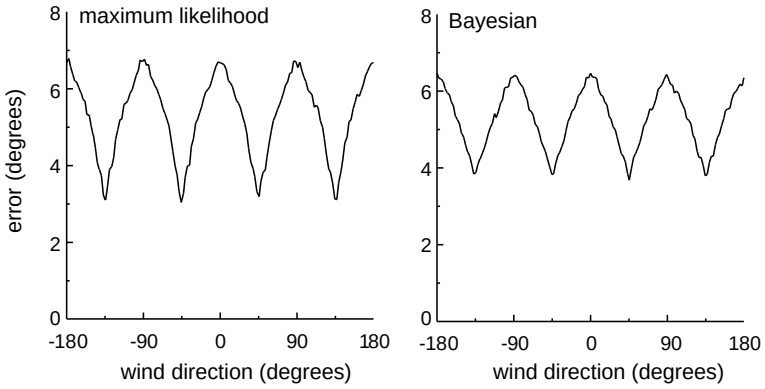

Figure 3.7 Maximum likelihood and Bayesian estimation errors for the cricket cercal system. ML and Bayesian estimates of the wind direction were compared with the actual stimulus value for a large number of simulated firing rates. Firing rates were generated as for figure 3.5B. The error shown is the root-mean-squared difference between the estimated and actual stimulus angles. (Adapted from Salinas and Abbott, 1994.)

ence between the estimate and the true value, $L(s, s_{\text{bayes}}) = (s - s_{\text{bayes}})^2$, the estimate that minimizes the expected loss is the mean

$$s_{\text{bayes}} = \int ds\, p[s|\mathbf{r}]s \ . \tag{3.27}$$

If the loss function is the absolute value of the difference, $L(s, s_{\text{bayes}}) = |s - s_{\text{bayes}}|$, then s_{bayes} is the median rather than the mean of the distribution $p[s|\mathbf{r}]$.

MAP inference

Maximum a posteriori (MAP) inference does not involve a loss function but instead simply chooses the stimulus value, s_{MAP}, that maximizes the conditional probability density of the stimulus, $p[s_{\text{MAP}}|\mathbf{r}]$. The MAP approach is thus to choose as the estimate s_{MAP} the most likely stimulus value for a given set of rates. If the prior or stimulus probability density $p[s]$ is independent of s, then $p[s|\mathbf{r}]$ and $p[\mathbf{r}|s]$ have the same dependence on s, because the factor $p[s]/p[\mathbf{r}]$ in equation 3.26 is independent of s. In this case, the MAP algorithm is equivalent to maximizing the likelihood function, that is, choosing s_{ML} to maximize $p[\mathbf{r}|s_{\text{ML}}]$, which is called maximum likelihood (ML) inference.

ML inference

Previously we applied the vector decoding method to the cercal system of the cricket. Figure 3.7 shows the root-mean-squared difference between the true and estimated wind directions for the cercal system, using ML and Bayesian methods. For the cercal interneurons, the response probability density $p[\mathbf{r}|s]$ is a product of four Gaussians with means and variances given by the data points and error bars in figure 3.4. The Bayesian estimate in figure 3.7 is based on the squared-difference loss function. Both estimates use a constant stimulus probability density $p[s]$, so the ML and MAP estimates are identical. The maximum likelihood estimate is either more or less accurate than the Bayesian estimate, depending on the angle.

monkey then has to release the resting key and press another one to get a fruit juice reward. The reward is delivered a short time after the second key is pressed. The upper plot of figure 9.3A shows the response of a cell in early trials. The cell responds vigorously to the reward, but only fires a little above baseline in response to the sound. The lower plot shows the response after a moderate amount of training. Now the cell responds to the sound, but not to the reward. The responses show a distinct similarity to the plots of $\delta(t)$ in figure 9.2.

The similarity between the responses of the dopaminergic neurons and the quantity $\delta(t)$ suggests that their activity provides a prediction error for reward, i.e. an ongoing difference between the amount of reward that is delivered and the amount that is expected. Figure 9.3B provides further evidence for this interpretation. It shows the activity of a dopamine cell in a task similar to that of figure 9.3A. The top row of this figure shows normal performance, and is just like the bottom row of figure 9.3A. The bottom row shows what happens when the monkey is expecting reward but it is not delivered. In this case, the cell's activity is inhibited below baseline at just the time it would have been activated by the reward in the original trials. This is in agreement with the prediction error interpretation of this activity.

Something similar to the temporal difference learning rule could be realized in a neural system if the dopamine signal representing δ acts to gate and regulate the plasticity associated with learning. We discuss this possibility further in a later section.

9.3 Static Action Choice

policy

In classical conditioning experiments, rewards are directly associated with stimuli. In more natural settings, rewards and punishments are associated with the actions an animal takes. Animals develop policies, or plans of action, that increase reward. In studying how this might be done, we consider two different cases. In static action choice, the reward or punishment immediately follows the action taken. In sequential action choice, rewards may be delayed until several actions are completed.

foraging

As an example of static action choice, we consider bees foraging among flowers in search of nectar. We model an experiment in which single bees forage under controlled conditions among blue and yellow artificial flowers (small dishes of sugar water sitting on colored cards). In actual experiments, the bees learn within a single session (involving visits to 40 artificial flowers) about the reward characteristics of the blue and yellow flowers. All else being equal, they preferentially land on the color of flower that delivers more reward. This preference is maintained over multiple sessions. However, if the reward characteristics of the flowers are interchanged, the bees quickly swap their preferences.

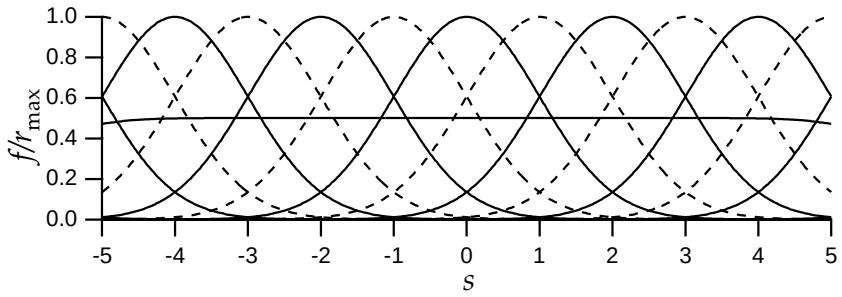

Figure 3.8 An array of Gaussian tuning curves spanning stimulus values from -5 to 5. The peak values of the tuning curves fall on the integer values of s and the tuning curves all have $\sigma_a = 1$. For clarity, the curves are drawn alternately with dashed and solid lines. The approximately flat curve with value near 0.5 is 1/5 the sum of the tuning curves shown, indicating that this sum is approximately independent of s.

bility for the population is the product of the individual probabilities,

$$P[\mathbf{r}|s] = \prod_{a=1}^{N} \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} \exp(-f_a(s)T) \,. \tag{3.30}$$

The assumption of independence simplifies the calculations considerably.

The filled circles in figure 3.9 show a set of randomly generated firing rates for the array of Gaussian tuning curves in figure 3.8 for $s = 0$. This figure also illustrates a useful way of visualizing population responses: plotting the responses as a function of the preferred stimulus values. The dashed curve in figure 3.9 is the tuning curve for the neuron with $s_a = 0$. Because the tuning curves are functions of $|s - s_a|$, the values of the dashed curve at $s_a = -5, -4, \ldots, 5$ are the mean activities of the cells with preferred values at those locations for a stimulus at $s = 0$.

To apply the ML estimation algorithm, we only need to consider the terms in $P[\mathbf{r}|s]$ that depend on s. Because equation 3.30 involves a product, it is convenient to take its logarithm and write

$$\ln P[\mathbf{r}|s] = T \sum_{a=1}^{N} r_a \ln\big(f_a(s)\big) + \ldots \,, \tag{3.31}$$

where the ellipsis represents terms that are independent or approximately independent of s, including, as discussed above, $\sum f_a(s)$. Because maximizing a function and maximizing its logarithm are equivalent, we can use the logarithm of the conditional probability in place of the actual probability in ML decoding.

The ML estimated stimulus, $s_{\rm ML}$, is the stimulus that maximizes the right side of equation 3.31. Setting the derivative to 0, we find that $s_{\rm ML}$ is deter-

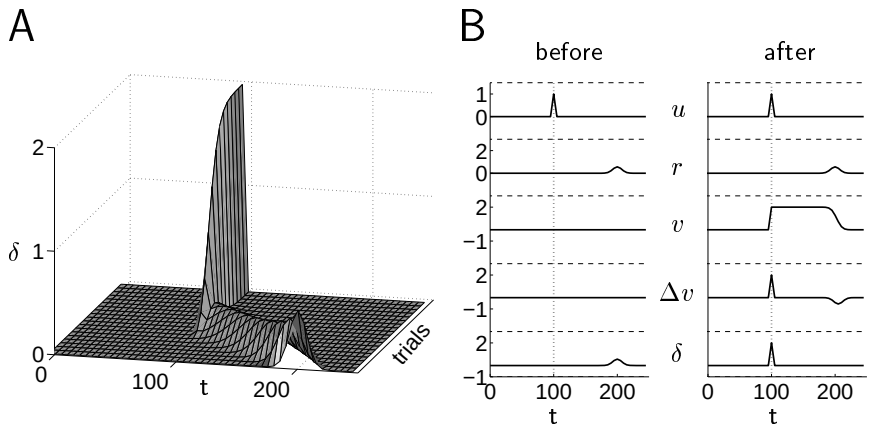

Figure 9.2 Learning to predict a reward. (A) The surface plot shows the prediction error $\delta(t)$ as a function of time within a trial, across trials. In the early trials, the peak error occurs at the time of the reward ($t = 200$), while in later trials it occurs at the time of the stimulus ($t = 100$). (B) The rows show the stimulus $u(t)$, the reward $r(t)$, the prediction $v(t)$, the temporal difference between predictions $\Delta v(t-1) = v(t) - v(t-1)$, and the full temporal difference error $\delta(t-1) = r(t-1) + \Delta v(t-1)$. The reward is presented over a short interval, and the prediction v sums the total reward. The left column shows the behavior before training, and the right column, after training. $\Delta v(t-1)$ and $\delta(t-1)$ are plotted instead of $\Delta v(t)$ and $\delta(t)$ because the latter quantities cannot be computed until time $t+1$, when $v(t+1)$ is available.

As the peak in δ moves backward from the time of the reward to the time of the stimulus, weights $w(\tau)$ for $\tau = 100, 99, \ldots$ successively grow. This gradually extends the prediction of future reward, $v(t)$, from an initial transient at the time of the reward to a broad plateau extending from the time of the stimulus to the time of the reward. Eventually, v predicts the correct total future reward from the time of the stimulus onward, and predicts the time of the reward delivery by dropping to 0 at the time when the reward is delivered. The exact shape of the ridge of activity that moves from $t = 200$ to $t = 100$ over the course of trials in figure 9.2A is sensitive to a number of factors, including the learning rate, and the form of the linear filter of equation 9.6.

Unlike the delta rule, the temporal difference rule provides an account of secondary conditioning. Suppose an association between stimulus s_1 and a future reward has been established, as in figure 9.2. When, as indicated in table 9.1, a second stimulus, s_2, is introduced before the first stimulus, the positive spike in $\delta(t)$ at the time that s_1 is presented drives an increase in the value of the weight associated with s_2, and thus establishes a positive association between the second stimulus and the reward. This exactly mirrors the primary learning process for s_1 described above. Of course, because the reward is not presented in these trials, there is a negative spike in $\delta(t)$ at the time of the reward itself, and ultimately the association between both s_1 and s_2 and the reward extinguishes.

The MAP estimation procedure is similar in spirit to the ML approach, but the MAP estimate, $s_{\rm MAP}$, may differ from $s_{\rm ML}$ if the probability density $p[s]$ depends on s. The MAP algorithm allows us to include prior knowledge about the distribution of stimulus values in the decoding estimate. As noted above, if the $p[s]$ is constant, the MAP and ML estimates are identical. In addition, if many neurons are observed, or if a small number of neurons is observed over a long trial period, even a nonconstant stimulus distribution has little effect and $s_{\rm MAP} \approx s_{\rm ML}$.

The MAP estimate is computed from the distribution $p[s|\mathbf{r}]$ determined by Bayes theorem. In terms of the logarithms of the probabilities, $\ln p[s|\mathbf{r}] = \ln P[\mathbf{r}|s] + \ln p[s] - \ln P[\mathbf{r}]$. The last term in this expression is independent of s and can be absorbed into the ignored s-independent terms, so we can write, as in equation 3.31,

$$\ln p[s|\mathbf{r}] = T\sum_{a=1}^{N} r_a \ln\big(f_a(s)\big) + \ln p[s] + \ldots . \tag{3.35}$$

Maximizing this determines the MAP estimate,

$$T\sum_{a=1}^{N} \frac{r_a f_a'(s_{\rm MAP})}{f_a(s_{\rm MAP})} + \frac{p'[s_{\rm MAP}]}{p[s_{\rm MAP}]} = 0\,. \tag{3.36}$$

If the stimulus or prior distribution is itself Gaussian with mean $s_{\rm prior}$ and variance $\sigma_{\rm prior}$, and we use the Gaussian array of tuning curves, equation 3.36 yields

$$s_{\rm MAP} = \frac{T\sum r_a s_a/\sigma_a^2 + s_{\rm prior}/\sigma_{\rm prior}^2}{T\sum r_a/\sigma_a^2 + 1/\sigma_{\rm prior}^2}\,. \tag{3.37}$$

Figure 3.10 compares the conditional stimulus probability densities $p[s|\mathbf{r}]$ for a constant stimulus distribution (solid curve) and for a Gaussian stimulus distribution with $s_{\rm prior} = -2$ and $\sigma_{\rm prior} = 1$, using the firing rates given by the filled circles in figure 3.9. If the stimulus distribution is constant, $p[s|\mathbf{r}]$ peaks near the true stimulus value of 0. The effect of a nonconstant stimulus distribution is to shift the curve toward the value -2, where the stimulus probability density has its maximum, and to decrease its width by a small amount. The estimate is shifted to the left because the prior distribution suggests that the stimulus is more likely to take negative values than positive ones, independent of the evoked response. The decreased width is due to the added information that the prior distribution provides. The curves in figure 3.10 can be computed from equations 3.28 and 3.35 as Gaussians with variances $1/(T\sum r_a/\sigma_a^2)$ (constant prior) and $1/(T\sum r_a/\sigma_a^2 + 1/\sigma_{\rm prior}^2)$ (Gaussian prior).

The accuracy with which an estimate $s_{\rm est}$ describes a stimulus s can be characterized by two important quantities, its bias $b_{\rm est}(s)$ and its variance $\sigma_{\rm est}^2(s)$. The bias is the difference between the average of $s_{\rm est}$ across trials that use the stimulus s and the true value of the stimulus (i.e., s), *bias*

$$b_{\rm est}(s) = \langle s_{\rm est}\rangle - s\,. \tag{3.38}$$

secondary conditioning

Secondary conditioning involves the association of one stimulus with a reward, followed by an association of a second stimulus with the first stimulus (table 9.1). This causes the second stimulus to evoke expectation of a reward with which it has never been paired (although if pairings of the two stimuli without the reward are repeated too many times, the result is extinction of the association of both stimuli with the reward). The delta rule cannot account for the positive expectation associated with the second stimulus. Indeed, because the reward does not appear when the second stimulus is presented, the delta rule would cause w_2 to become negative. In other words, in this case, the delta rule would predict inhibitory, not excitatory, secondary conditioning. Secondary conditioning is related to the problem of delayed rewards in instrumental conditioning that we discuss later in this chapter.

Secondary conditioning raises the important issue of keeping track of the time within a trial in which stimuli and rewards are present. This is evident because a positive association with the second stimulus is reliably established only if it precedes the first stimulus in the trials in which they are paired. If the two stimuli are presented simultaneously, the result may be inhibitory rather than secondary conditioning.

Predicting Future Reward: Temporal Difference Learning

We measure time within a trial using a discrete time variable t, which falls in the range $0 \leq t \leq T$. The stimulus $u(t)$, the prediction $v(t)$, and the reward $r(t)$ are all expressed as functions of t.

In addition to associating stimuli with rewards and punishments, animals can learn to predict the future time within a trial at which reinforcement will be delivered. We might therefore be tempted to interpret $v(t)$ as the reward predicted to be delivered at time step t. However, Sutton and Barto (1990) suggested an alternative interpretation of $v(t)$ that provides a better match to psychological and neurobiological data, and suggests how animals might use their predictions to optimize behavior when rewards are delayed. The suggestion is that the variable $v(t)$ should be interpreted as a prediction of the total future reward expected from time t onward to the end of the trial, namely

total future reward

$$\left\langle \sum_{\tau=0}^{T-t} r(t+\tau) \right\rangle . \tag{9.5}$$

The brackets denote an average over trials. This quantity is useful for optimization, because it summarizes the total expected worth of the current state. To approximate $v(t)$, we generalize the linear relationship used for classical conditioning, equation 9.3. For the case of a single time-dependent stimulus $u(t)$, we write

$$v(t) = \sum_{\tau=0}^{t} w(\tau) u(t-\tau) . \tag{9.6}$$

to (appendix B)

$$\sigma^2_{\text{est}}(s) \geq \frac{\left(1 + b'_{\text{est}}(s)\right)^2}{I_{\text{F}}(s)}, \tag{3.41}$$

where $b'_{\text{est}}(s)$ is the derivative of $b_{\text{est}}(s)$. If we assume here that the firing rates take continuous values and that their distribution in response to a stimulus s is described by the conditional probability density $p[\mathbf{r}|s]$, the quantity $I_{\text{F}}(s)$ in equation 3.41 is the Fisher information of the firing-rate distribution, which is related to $p[\mathbf{r}|s]$ (assuming the latter is sufficiently smooth) by

Fisher information

$$I_{\text{F}}(s) = \left\langle -\frac{\partial^2 \ln p[\mathbf{r}|s]}{\partial s^2} \right\rangle = \int d\mathbf{r}\, p[\mathbf{r}|s] \left(-\frac{\partial^2 \ln p[\mathbf{r}|s]}{\partial s^2} \right). \tag{3.42}$$

The reader can verify that the Fisher information can also be written as

$$I_{\text{F}}(s) = \left\langle \left(\frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \right)^2 \right\rangle = \int d\mathbf{r}\, p[\mathbf{r}|s] \left(\frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \right)^2. \tag{3.43}$$

The Cramér-Rao bound sets a limit on the accuracy of any unbiased estimate of the stimulus. When $b_{\text{est}}(s) = 0$, equation 3.40 indicates that the average squared estimation error is equal to σ^2_{est} and, by equation 3.41, this satisfies the bound $\sigma^2_{\text{est}} \geq 1/I_{\text{F}}(s)$. Provided that we restrict ourselves to unbiased decoding schemes, the Fisher information sets an absolute limit on decoding accuracy, and it thus provides a useful limit on encoding accuracy. Although imposing zero bias on the decoding estimate seems reasonable, the restriction is not trivial. In general, minimizing the decoding error in equation 3.40 involves a trade-off between minimizing the bias and minimizing the variance of the estimator. In some cases, biased schemes may produce more accurate results than unbiased ones. For a biased estimator, the average squared estimation error and the variance of the estimate are not equal, and the estimation error can be either larger or smaller than $1/I_{\text{F}}(s)$.

The limit on decoding accuracy set by the Fisher information can be attained by a decoding scheme we have studied, the maximum likelihood method. In the limit of large numbers of encoding neurons, and for most firing-rate distributions, the ML estimate is unbiased and saturates the Cramér-Rao bound. In other words, the variance of the ML estimate is given asymptotically (for large N) by $\sigma^2_{\text{ML}}(s) = 1/I_{\text{F}}(s)$. Any unbiased estimator that saturates the Cramér-Rao lower bound is called efficient.

efficiency

Furthermore, $I_{\text{F}}(s)$ grows linearly with N, and the ML estimate obeys a central limit theorem, so that $N^{1/2}(s_{\text{ML}} - s)$ is Gaussian distributed with a variance that is independent of N in the large N limit. Finally, in the limit $N \to \infty$, the ML estimate is asymptotically consistent, in the sense that $P[|s_{\text{ML}} - s| > \epsilon] \to 0$ for any $\epsilon > 0$.

asymptotic consistency

As equation 3.42 shows, the Fisher information is a measure of the expected curvature of the log likelihood at stimulus value s. Curvature is

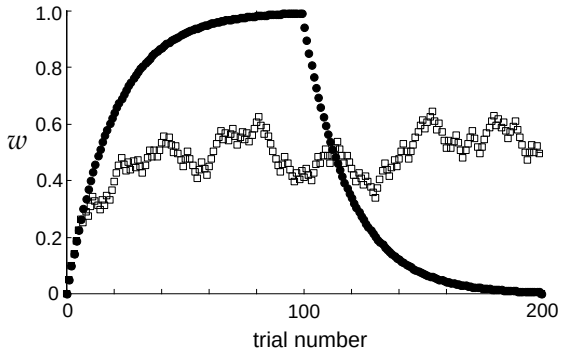

Figure 9.1 Acquisition and extinction curves for Pavlovian conditioning and partial reinforcement as predicted by the Rescorla-Wagner model. The filled circles show the time evolution of the weight w over 200 trials. In the first 100 trials, a reward of $r = 1$ was paired with the stimulus, while in trials 100-200 no reward was paired ($r = 0$). Open squares show the evolution of the weights when a reward of $r = 1$ was paired with the stimulus randomly on 50% of the trials. In both cases, $\epsilon = 0.05$.

curves are generally more sigmoidal in shape. There are various ways to account for this discrepancy, the simplest of which is to assume a nonlinear relationship between the expectation v and the behavior of the animal.

partial reinforcement

The Rescorla-Wagner rule also accounts for aspects of the phenomenon of partial reinforcement, in which a reward is associated with a stimulus only on a random fraction of trials (table 9.1). Behavioral measures of the ultimate association of the reward with the stimulus in these cases indicate that it is weaker than when the reward is always presented. This is expected from the delta rule, because the ultimate steady-state average value of $w = \langle r \rangle$ is smaller than r in this case. The open squares in figure 9.1 show what happens to the weight when the reward is associated with the stimulus 50% of the time. After an initial rise from 0, the weight varies randomly around an average value of 0.5.

stimulus vector $\mathbf{u}$
weight vector $\mathbf{w}$

To account for experiments in which more than one stimulus is used in association with a reward, the Rescorla-Wagner rule must be extended to include multiple stimuli. This is done by introducing a vector of binary variables $\mathbf{u}$, with each of its components representing the presence or absence of a given stimulus, together with a vector of weights $\mathbf{w}$. The expected reward is then the sum of each stimulus parameter multiplied by its corresponding weight, written compactly as a dot product,

$$v = \mathbf{w} \cdot \mathbf{u} \ . \tag{9.3}$$

delta rule

Minimizing the prediction error by stochastic gradient descent in this case gives the delta learning rule,

$$\mathbf{w} \to \mathbf{w} + \epsilon \delta \mathbf{u} \quad \text{with} \quad \delta = r - v \, . \tag{9.4}$$

blocking

Various classical conditioning experiments probe the way that predictions are shared between multiple stimuli (see table 9.1). Blocking is the

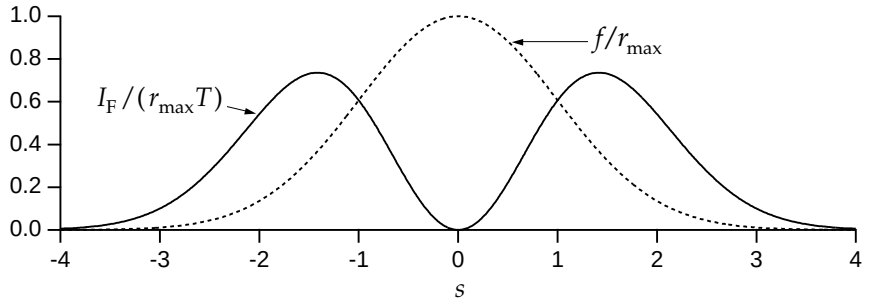

Figure 3.11 The Fisher information for a single neuron with a Gaussian tuning curve with $s = 0$ and $\sigma_a = 1$, and Poisson variability. The Fisher information (solid curve) has been divided by $r_{max}T$, the peak firing rate of the tuning curve times the duration of the trial. The dashed curve shows the tuning curve scaled by r_{max}. Note that the Fisher information is greatest where the slope of the tuning curve is highest, and vanishes at $s = 0$, where the tuning curve peaks.

rons with tuning curves of identical shapes, distributed evenly over a range of stimulus values as in figure 3.8. Equation 3.45 indicates that the Fisher information will be largest if the tuning curves of individual neurons are rapidly varying (making the square of their derivatives large), and if many neurons respond (making the sum over neurons large). For typical neuronal response tuning curves, these two requirements are in conflict with one another. If the population of neurons has narrow tuning curves, individual neural responses are rapidly varying functions of the stimulus, but few neurons respond. Broad tuning curves allow many neurons to respond, but the individual responses are not as sensitive to the stimulus value. To determine whether narrow or broad tuning curves produce the more accurate encodings, we consider a dense distribution of Gaussian tuning curves, all with $\sigma_a = \sigma_r$. Using such curves in equation 3.45, we find

$$I_F(s) = T\sum_{a=1}^{N} \frac{r_{max}(s - s_a)^2}{\sigma_r^4} \exp\left(-\frac{1}{2}\left(\frac{s - s_a}{\sigma_r}\right)^2\right). \qquad (3.46)$$

This expression can be approximated by replacing the sum over neurons with an integral over their preferred stimulus values and multiplying by a density factor ρ_s. The factor ρ_s is the density with which the neurons cover the range of stimulus values, and it is equal to the number of neurons with preferred stimulus values lying within a unit range of s values. Replacing the sum over a with an integral over a continuous preferred stimulus parameter ξ (which replaces s_a), we find

sums→integrals

$$\begin{aligned} I_F(s) &\approx \rho_s T \int_{-\infty}^{\infty} d\xi \frac{r_{max}(s - \xi)^2}{\sigma_r^4} \exp\left(-\frac{1}{2}\left(\frac{s - \xi}{\sigma_r}\right)^2\right) \\ &= \frac{\sqrt{2\pi}\rho_s \sigma_r r_{max} T}{\sigma_r^2}. \end{aligned} \qquad (3.47)$$

We have expressed the final result in this form because the number of neurons that respond to a given stimulus value is roughly $\rho_s\sigma_r$, and the Fisher

delayed rewards

until the sequence is completed. Thus, learning the appropriate action at each step in the sequence must be based on future expectation, rather than immediate receipt, of reward. This makes learning more difficult. Despite the differences between classical and instrumental conditioning, we show how to use the temporal difference model we discuss for classical conditioning as the heart of a model of instrumental conditioning when rewards are delayed.

For consistency with the literature on reinforcement learning, throughout this chapter, the letter r is used to represent a reward rather than a firing rate. Also, for convenience, we consider discrete actions such as a choice between two alternatives, rather than a continuous range of actions. We also consider trials that consist of a number of discrete events and use an integer time variable $t = 0, 1, 2, \ldots$ to indicate steps during a trial. We therefore also use discrete weight update rules (like those we discussed for supervised learning in chapter 8) rather than learning rules described by differential equations.

9.2 Classical Conditioning

Classical conditioning involves a wide range of different training and testing procedures and a rich set of behavioral phenomena. The basic procedures and results we discuss are summarized in table 9.1. Rather than going through the entries in the table at this point, we introduce a learning algorithm that serves to summarize and structure these results.

unconditioned stimulus and response

conditioned stimulus and response

In the classic Pavlovian experiment, dogs are repeatedly fed just after a bell is rung. Subsequently, the dogs salivate whenever the bell sounds, as if they expect food to arrive. The food is called the unconditioned stimulus. Dogs naturally salivate when they receive food, and salivation is thus called the unconditioned response. The bell is called the conditioned stimulus because it elicits salivation only under the condition that there has been prior learning. The learned salivary response to the bell is called the conditioned response. We do not use this terminology in the following discussion. Instead, we treat those aspects of the conditioned responses that mark the animal's expectation of the delivery of reward, and build models of how these expectations are learned. We therefore refer to stimuli, rewards, and expectation of reward.

Predicting Reward: The Rescorla-Wagner Rule

The Rescorla-Wagner rule (Rescorla and Wagner, 1972), which is a version of the delta rule of chapter 8, provides a concise account of certain aspects of classical conditioning. The rule is based on a simple linear prediction of the reward associated with a stimulus. We use a binary variable u to represent the presence or absence of the stimulus ($u = 1$ if the stimulus

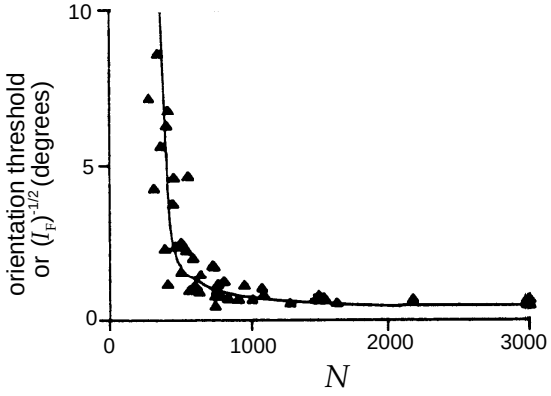

Figure 3.12 Comparison of Fisher information and discrimination thresholds for orientation tuning. The solid curve is the minimum standard deviation of an estimate of orientation angle from the Cramér-Rao bound, plotted as a function of the number of neurons (N) involved in the estimation. The triangles are data points from an experiment that determined the threshold for discrimination of the orientation of line images by human subjects as a function of line length and eccentricity. An effective number of neurons involved in the task was estimated for the different line lengths and eccentricities, using the cortical magnification factor discussed in chapter 2. (Adapted from Paradiso, 1988.)

Figure 3.12 shows an interesting comparison of the Fisher information for orientation tuning in the primary visual cortex with human orientation discrimination thresholds. Agreement like this can occur for difficult tasks, like discrimination at threshold, where the performance of a subject may be limited by basic constraints on neuronal encoding accuracy.

3.4 Spike-Train Decoding

The decoding methods we have considered estimate or discriminate static stimulus values on the basis of spike-count firing rates. Spike-count firing rates do not provide sufficient information for reconstructing a stimulus that varies during the course of a trial. Instead, we can estimate such a stimulus from the sequence of firing times t_i for $i = 1, 2, \ldots, n$ of the spikes that it evokes. One method for doing this is similar to the Wiener kernel approach used to estimate the firing rate from the stimulus in chapter 2, and to approximate a firing rate using a sliding window function in chapter 1. For simplicity, we restrict our discussion to the decoding of a single neuron. We assume, as we did in chapter 2, that the time average of the stimulus being estimated is 0.

In spike-train decoding, we attempt to construct an estimate of the stimulus at time t from the sequence of spikes evoked up to that time. There are paradoxical aspects of this procedure. The firing of an action potential at time t_i is only affected by the stimulus $s(t)$ prior to that time, $t < t_i$, and yet, in spike decoding, we attempt to extract information from this

spike occurring at time t_i contributes a kernel $K(t - t_i)$, and the total estimate is obtained by summing over all spikes,

$$s_{\rm est}(t - \tau_0) = \sum_{i=1}^{n} K(t - t_i) - \langle r \rangle \int_{-\infty}^{\infty} d\tau\, K(\tau)\,. \tag{3.51}$$

The last term, with $\langle r \rangle = \langle n \rangle / T$ the average firing rate over the trial, is included to impose the condition that the time average of $s_{\rm est}$ is 0, in agreement with the time-average condition on s. The sum in equation 3.51 includes all spikes, so the constraint that only those spikes occurring prior to the time t (spikes 1-7 in figure 3.13A) should be included must be imposed by requiring $K(t - t_i) = 0$ for $t - t_i \leq 0$. A kernel satisfying this constraint is termed causal. We ignore the causality constraint for now and construct an acausal kernel, but we will return to issues of causality later in the discussion. Figure 3.13A shows how spikes contribute to a stimulus estimate, using the kernel shown in figure 3.13B.

Equation 3.51 can be written in a compact way by using the neural response function $\rho(t) = \sum \delta(t - t_i)$ introduced in chapter 1,

$$s_{\rm est}(t - \tau_0) = \int_{-\infty}^{\infty} d\tau\ (\rho(t - \tau) - \langle r \rangle)\, K(\tau)\,. \tag{3.52}$$

Using this form of the estimate, the construction of the optimal kernel K proceeds very much like the construction of the optimal kernel for predicting firing rates in chapter 2. We choose K so that the squared difference between the estimated stimulus and the actual stimulus, averaged over both time and trials,

$$\frac{1}{T}\int_0^T dt \left\langle \left(\int_{-\infty}^{\infty} d\tau\ (\rho(t - \tau) - \langle r \rangle)\, K(\tau) - s(t - \tau_0) \right)^2 \right\rangle, \tag{3.53}$$

is minimized. The calculation proceeds as in appendix A of chapter 2, and the result is that K obeys the equation

$$\int_{-\infty}^{\infty} d\tau'\, Q_{\rho\rho}(\tau - \tau') K(\tau') = Q_{rs}(\tau - \tau_0)\,, \tag{3.54}$$

where $Q_{\rho\rho}$ is the spike-train autocorrelation function,

$$Q_{\rho\rho}(\tau - \tau') = \frac{1}{T}\int_0^T dt \left\langle (\rho(t - \tau) - \langle r \rangle)(\rho(t - \tau') - \langle r \rangle) \right\rangle, \tag{3.55}$$

as defined in chapter 1. Q_{rs} is the correlation of the firing rate and the stimulus, which is related to the spike-triggered average C, both introduced in chapter 1,

$$Q_{rs}(\tau - \tau_0) = \langle r \rangle C(\tau_0 - \tau) = \frac{1}{T}\left\langle \sum_{i=1}^{n} s(t_i + \tau - \tau_0) \right\rangle. \tag{3.56}$$

At this point in the derivation of the optimal linear kernel for firing-rate prediction in chapter 2, we chose the stimulus to be uncorrelated so that an

change, we find that

$$\psi(\mathbf{w}, \gamma) \geq n\delta . \tag{8.76}$$

Similarly, over one learning step in which some change is made,

$$\Delta|\mathbf{w}, \gamma|^2 = 2(\mathbf{w} \cdot \mathbf{u}^m - \gamma)v^m + |\mathbf{u}^m|^2 + 1 . \tag{8.77}$$

The first term on the right side is always negative when an error is made, and if we define D to be the maximum value of $|\mathbf{u}^m|^2$ over all the training samples, we find

$$\Delta|\mathbf{w}, \gamma|^2 < D + 1 . \tag{8.78}$$

After n nontrivial learning iterations (iterations in which the weights and threshold are modified) starting from $|\mathbf{w}, \gamma|^2 = 0$, we therefore have

$$|\mathbf{w}, \gamma|^2 < n(D + 1) \tag{8.79}$$

Putting together equations 8.76 and 8.79, we find, after n nontrivial training steps,

$$\Phi(\mathbf{w}, \gamma) > \frac{n\delta}{\sqrt{n(D+1)}} . \tag{8.80}$$

To ensure that $\Phi(\mathbf{w}, \gamma) \leq 1$, we must have $n \leq (D + 1)/\delta^2$. Therefore, after a finite number of weight changes, the perceptron learning rule must stop changing the weights, and the perceptron must classify all the patterns correctly (although the weights and threshold that result are not necessarily equal to $\mathbf{w}^*$ and γ^*).

8.7 Annotated Bibliography

Hebb's (1949) original proposal about learning set the stage for many of the subsequent investigations. We followed the treatments of Hebbian, BCM, anti-Hebbian, and trace learning found in Goodall (1960), Sejnowski (1977), Bienenstock et al. (1982), Oja (1982), Földiák (1989; 1991), Leen (1991), Atick & Redlich (1993), and Wallis & Baddeley (1997). Extensive coverage of these topics and related analyses can be found in **Hertz et al. (1991)**. We followed Miller & MacKay (1994) and Miller (1996b) in the analysis of weight constraints and normalization. Jolliffe (1986) treats principal components analysis theoretically (see also chapter 10). Intrator & Cooper (1992) considers the BCM rule from the statistical perspective of projection pursuit (Huber, 1985).

Sejnowski (1999) comments on the relationship between Hebb's suggestions and recent experimental data and theoretical studies on temporal sensitivity in Hebbian plasticity (see Minai & Levy, 1993; Blum & Abbott,

small for negative values of τ by choosing τ_0 large enough, but this may require a fairly large prediction delay. We can force exact adherence to the causality constraint for $\tau < 0$ by replacing $K(\tau)$ by $\Theta(\tau)K(\tau)$, where $\Theta(\tau)$ is defined such that $\Theta(\tau) = 1$ for $\tau > 0$ and $\Theta(\tau) = 0$ for $\tau < 0$. The causality constraint was imposed in this way in figure 3.13B. When it is multiplied by $\Theta(\tau)$, the restricted K is no longer the optimal decoding kernel, but it may be close to optimal.

causality constraint

Another way of imposing causality on the decoding kernel is to expand $K(\tau)$ as a weighted sum of causal basis functions (functions that vanish for negative arguments and span the space of functions satisfying the causal constraint). The optimal weights are then determined by minimizing the estimation error. This approach has the advantage of producing a truly optimal kernel for any desired value of τ_0. A simpler but nonoptimal approach is to consider a fixed functional form for $K(\tau)$ that vanishes for $\tau \leq 0$ and is characterized by a number of free parameters that can be determined by minimizing the decoding error. Finally, the optimal causal kernel, also called the Wiener-Hopf filter, can be obtained by a technique that involves so-called spectral factorization of $\tilde{Q}_{\rho\rho}(\omega)$.

Figure 3.14 shows an example of spike-train decoding for the H1 neuron of the fly discussed in chapter 2. The top panel gives two reconstruction kernels, one acausal and one causal, and the bottom panel compares the reconstructed stimulus velocity with the actual stimulus velocity. The middle panel in figure 3.14 points out one further wrinkle in the procedure. Flies have two H1 neurons, one on each side of the body, that respond to motion in opposite directions. As is often the case, half-wave rectification prevents a single neuron from encoding both directions of motion. In the experiment described in the figure, rather than recording from both H1 neurons, Bialek et al. (1991) recorded from a single H1 neuron, but presented both the stimulus $s(t)$ and its negative, $-s(t)$. The two rows of spikes in the middle panel show sample traces for each of these presentations. This procedure provides a reasonable approximation of recording both H1 neurons, and produces a neuron/anti-neuron pair of recordings similar to the one that we discussed in connection with motion discrimination from area MT neurons. The stimulus is then decoded by summing the kernel $K(t - t_i)$ for all spike times t_i of the recorded H1 neuron and summing $-K(t - t_j)$ for all spike times t_j of its anti-neuron partner.

The fly has only two H1 neurons from which it must extract information about visual motion, so it seems reasonable that stimulus reconstruction using the spike-train decoding technique can produce quite accurate results (figure 3.14). It is perhaps more surprising that accurate decoding, at least in the sense of percent correct discriminations, can be obtained from single neurons out of the large population of MT neurons responding to visual motion in the monkey. Of course, the reconstruction of a time-dependent stimulus from H1 responses is more challenging than the binary discrimination done with MT neurons. Furthermore, it is worth remembering that in all the examples we have considered, including decoding wind direction from the cercal system and arm movement direction

The derivation of an unsupervised learning rule for this Boltzmann machine proceeds very much like the derivation we presented for the supervised case. The equivalent of equation 8.65 is

$$\frac{\partial \ln P[\mathbf{u}^m; \mathbf{W}]}{\partial W_{ab}} = \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}^m; \mathbf{W}] v_a u_b^m - \sum_{\mathbf{u},\mathbf{v}} P[\mathbf{u}, \mathbf{v}; \mathbf{W}] v_a u_b \, . \qquad (8.71)$$

In this case, both terms must be evaluated by Gibbs sampling. The wake phase Hebbian term requires a stochastic output $\mathbf{v}(\mathbf{u}^m)$, which is calculated from the sample input $\mathbf{u}^m$, just as it was for the anti-Hebbian term in equation 8.66. However, the sleep phase anti-Hebbian term in this case requires both an input $\mathbf{u}$ and an output $\mathbf{v}$ generated by the network. These are computed using a Gibbs sampling procedure in which both input and output states are stochastically generated through repeated Gibbs sampling. A randomly chosen component v_a is set to 1 with probability $F(\sum_b W_{ab} u_b)$ (or 0 otherwise), and a random component u_b is set to 1 with probability $F(\sum_a v_a W_{ab})$ (or 0 otherwise). Note that this corresponds to having the input units drive the output units in a feedforward manner through the weights $\mathbf{W}$, and having the output units drive the input units in a reversed manner using feedback weights with the same values. Once the network has settled to equilibrium through repeated Gibbs sampling of this sort, and the stochastic inputs and outputs have been generated, the full learning rule is

unsupervised learning for $\mathbf{W}$

$$W_{ab} \rightarrow W_{ab} + \epsilon_w \left(v_a(\mathbf{u}^m) u_b^m - v_a u_b \right) \, . \qquad (8.72)$$

The unsupervised learning rule can be extended to include recurrent connections by following the same general procedure.

8.5 Chapter Summary

We presented a variety of forms of Hebbian synaptic plasticity, ranging from the basic Hebb rule to rules that involve multiplicative and subtractive normalization, constant or sliding thresholds, and spike-timing effects. Two important features, stability and competition, were emphasized. We showed how the effects of unsupervised Hebbian learning could be estimated by computing the principal eigenvector of the correlation matrix of the inputs used during training. Unsupervised Hebbian learning can be interpreted as a process that produces weights that project the input vector onto the direction of maximal variance in the training data set. In some cases, this requires an extension from correlation-based to covariance-based rules. We used the principal eigenvector approach to analyze Hebbian models of the development of ocular dominance and its associated map in primary visual cortex. Plasticity rules based on the dependence of synaptic modification on spike timing were shown to implement temporal sequence and trace learning.

Forcing multiple outputs to have different selectivities requires them to be connected, either through fixed weights or by weights that are themselves

tio tests, and the Neyman-Pearson lemma. For static parameter decoding we introduced the vector method; Bayesian, maximum a posteriori, and maximum likelihood inference; the Fisher information; and the Cramér-Rao lower bound. We also showed how to use ideas from Wiener filtering to reconstruct an approximation of a time-varying stimulus from the spike trains it evokes.

3.6 Appendices

A: The Neyman-Pearson Lemma

Consider the difference $\Delta\beta$ in the power of two tests that have identical sizes α. One uses the likelihood ratio $l(r)$, and the other uses a different test function $h(r)$. For the test $h(r)$ using the threshold z_h,

$$\alpha_h(z_h) = \int dr\, p[r|-]\Theta(h(r)-z_h) \text{ and } \beta_h(z_h) = \int dr\, p[r|+]\Theta(h(r)-z_h)\,. \tag{3.61}$$

Similar equations hold for the $\alpha_l(z_l)$ and $\beta_l(z_l)$ values for the test $l(r)$ using the threshold z_l. We use the Θ function, which is 1 for positive and 0 for negative values of its argument, to impose the condition that the test is greater than the threshold. Comparing the β values for the two tests, we find

$$\Delta\beta = \beta_l(z_l) - \beta_h(z_h) = \int dr\, p[r|+]\Theta(l(r)-z_l) - \int dr\, p[r|+]\Theta(h(r)-z_h)\,. \tag{3.62}$$

The range of integration where $l(r) \geq z_l$ and also $h(r) \geq z_h$ cancels between these two integrals, so, in a more compact notation, we can write

$$\Delta\beta = \int dr\, p[r|+]\left(\Theta(l(r)-z_l)\Theta(z_h-h(r)) - \Theta(z_l-l(r))\Theta(h(r)-z_h)\right)\,. \tag{3.63}$$

Using the definition $l(r) = p[r|+]/p[r|-]$, we can replace $p[r|+]$ with $l(r)p[r|-]$ in this equation, giving

$$\Delta\beta = \int dr\, l(r)p[r|-]\Big(\Theta(l(r)-z_l)\Theta(z_h-h(r)) - \Theta(z_l-l(r))\Theta(h(r)-z_h)\Big)\,. \tag{3.64}$$

Then, due to the conditions imposed on $l(r)$ by the Θ functions within the integrals, replacing $l(r)$ by z can neither decrease the value of the integral resulting from the first term in the large parentheses, nor increase the value arising from the second. This leads to the inequality

$$\Delta\beta \geq z \int dr\, p[r|-]\left(\Theta(l(r)-z_l)\Theta(z_h-h(r)) - \Theta(z_l-l(r))\Theta(h(r)-z_h)\right)\,. \tag{3.65}$$

As in the discussion of the delta rule, it is more convenient to use a stochastic gradient ascent rule, choosing an index m at random to provide a Monte Carlo sample from the average of equation 8.64, and changing W_{ab} according to the derivative with respect to this sample,

$$\frac{\partial \ln P[\mathbf{v}^m|\mathbf{u}^m;\mathbf{W}]}{\partial W_{ab}} = \frac{\partial}{\partial W_{ab}}\Big(-E(\mathbf{u}^m,\mathbf{v}^m) - \ln Z(\mathbf{u}^m)\Big)$$
$$= v_a^m u_b^m - \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}] v_a u_b^m \,. \tag{8.65}$$

This derivative has a simple form for the Boltzmann machine because of equation 8.62.

Before we derive the stochastic gradient ascent learning rule, we need to evaluate the sum over $\mathbf{v}$ in the last term of the bottom line of equation 8.65. For Boltzmann machines with recurrent connections, like the ones we discuss below, this average cannot be calculated tractably. However, it can be estimated by stochastic sampling. In other words, we approximate the average over $\mathbf{v}$ by a single instance of a particular output $\mathbf{v}(\mathbf{u}^m)$ generated by the Boltzmann machine in response to the input $\mathbf{u}^m$. Making this replacement and setting the change in the weight matrix proportional to the derivative in equation 8.65, we obtain the learning rule

supervised learning for $\mathbf{W}$

$$W_{ab} \rightarrow W_{ab} + \epsilon_w \left(v_a^m u_b^m - v_a(\mathbf{u}^m) u_b^m\right) \,. \tag{8.66}$$

Equation 8.66 is identical in form to the perceptron learning rule of equation 8.56, except that $\mathbf{v}(\mathbf{u}^m)$ is computed from the input $\mathbf{u}^m$ by Gibbs sampling rather than by a deterministic rule. As discussed at the end of the previous section, equation 8.66 can also be interpreted as the difference of Hebbian and anti-Hebbian terms. The Hebbian term $v_a^m u_b^m$ is based on the sample input $\mathbf{u}^m$ and output $\mathbf{v}^m$. The anti-Hebbian term $-v_a(\mathbf{u}^m)u_b^m$ involves the sample input $\mathbf{u}^m$ and an output $\mathbf{v}(\mathbf{u}^m)$ generated by the Boltzmann machine in response to this input, rather than the sample output $\mathbf{v}^m$. In other words, whereas $\mathbf{v}^m$ is provided externally, $\mathbf{v}(\mathbf{u}^m)$ is obtained by Gibbs sampling, using the input $\mathbf{u}^m$ and the current values of the network weights. The overall learning rule is called a contrastive Hebb rule because it depends on the difference between Hebbian and anti-Hebbian terms. W_{ab} stops changing when the average of $v_a(\mathbf{u}^m)u_b^m$ over the input samples and network outputs equals the average of $v_a^m u_b^m$ over the input and output samples.

contrastive Hebb rule

Supervised learning for the Boltzmann machine is run in two phases, both of which use a sample input $\mathbf{u}^m$. The first phase, sometimes called the wake phase, involves Hebbian plasticity between sample inputs and outputs. The dynamics of the Boltzmann machine play no role during this phase. The second phase, called the sleep phase, consists of the network "dreaming" (i.e., internally generating) $\mathbf{v}(\mathbf{u}^m)$ in response to $\mathbf{u}^m$ based on the current weights $\mathbf{W}$. Then, anti-Hebbian learning based on $\mathbf{u}^m$ and $\mathbf{v}(\mathbf{u}^m)$ is applied to the weight matrix. Gibbs sampling is typically used to generate $\mathbf{v}(\mathbf{u}^m)$ from $\mathbf{u}^m$. It is also possible to use the mean-field method

wake phase

sleep phase

The last equality follows from the identity

$$\int d\mathbf{r}\, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \langle s_{\text{est}} \rangle = \langle s_{\text{est}} \rangle \int d\mathbf{r}\, \frac{\partial p[\mathbf{r}|s]}{\partial s} = 0 \tag{3.73}$$

because $\int d\mathbf{r}\, p[\mathbf{r}|s] = 1$. The last line of equation 3.72 is just another way of writing the expression being squared on the right side of the inequality 3.70, so combining this result with the inequality gives

$$\sigma^2_{\text{est}}(s) I_{\text{F}} \geq (1 + b'_{\text{est}}(s))^2 \,, \tag{3.74}$$

which, when rearranged, is the Cramér-Rao bound of equation 3.41.

C: The Optimal Spike-Decoding Filter

The optimal linear kernel for spike-train decoding is determined by solving equation 3.54. This is done by taking the Fourier transform of both sides of the equation, that is, multiplying both sides by $\exp(i\omega\tau)$ and integrating over τ,

$$\int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) \int_{-\infty}^{\infty} d\tau'\, Q_{\rho\rho}(\tau - \tau') K(\tau') = \int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) Q_{\text{rs}}(\tau - \tau_0) \,. \tag{3.75}$$

By making the replacement of integration variable $\tau \to \tau + \tau_0$, we find that the right side of this equation is

$$\exp(i\omega\tau_0) \int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) Q_{\text{rs}}(\tau) = \exp(i\omega\tau_0) \tilde{Q}_{\text{rs}}(\omega) \,, \tag{3.76}$$

where $\tilde{Q}_{\text{rs}}(\omega)$ is the Fourier transform of $Q_{\text{rs}}(\tau)$. The integral of the product of two functions that appears on the left side of equations 3.54 and 3.75 is a convolution. As a result of the theorem on the Fourier transforms of convolutions (see the Mathematical Appendix),

$$\int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) \int_{-\infty}^{\infty} d\tau'\, Q_{\rho\rho}(\tau - \tau') K(\tau') = \tilde{Q}_{\rho\rho}(\omega) \tilde{K}(\omega) \,, \tag{3.77}$$

where $\tilde{Q}_{\rho\rho}(\omega)$ and $\tilde{K}(\omega)$ are the Fourier transforms of $Q_{\rho\rho}(\tau)$ and $K(\tau)$ respectively:

$$\tilde{Q}_{\rho\rho}(\omega) = \int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) Q_{\rho\rho}(\tau) \quad \text{and} \quad \tilde{K}(\omega) = \int_{-\infty}^{\infty} d\tau\, \exp(i\omega\tau) K(\tau) \,. \tag{3.78}$$

Putting the left and right sides of equation 3.75 together as we have evaluated them, we find that

$$\tilde{Q}_{\rho\rho}(\omega) \tilde{K}(\omega) = \exp(i\omega\tau_0) \tilde{Q}_{\text{rs}}(\omega) \,. \tag{3.79}$$

Equation 3.60 follows directly from this result, and equation 3.59 then determines $K(\tau)$ as the inverse Fourier transform of $\tilde{K}(\omega)$.

Contrastive Hebbian Learning in Stochastic Networks

In chapter 7, we presented the Boltzmann machine, a stochastic network with binary units. One of the key innovations associated with the Boltzmann machine is a synaptic modification rule that has a sound foundation in probability theory. We start by describing the case of supervised learning, although the underlying theory is similar for both supervised and unsupervised cases with the Boltzmann machine.

Recall from chapter 7 that the Boltzmann machine produces a stochastic output $\mathbf{v}$ from an input $\mathbf{u}$ through a process called Gibbs sampling. This has two main consequences. First, rather than being described by an equation such as 8.2, the input-output relationship of a stochastic network is described by a distribution $P[\mathbf{v}|\mathbf{u}; \mathbf{W}]$, which is the probability that input $\mathbf{u}$ generates output $\mathbf{v}$ when the weight matrix of the network is $\mathbf{W}$. Second, supervised learning in a deterministic network involves the development of an input-output relationship that matches, as closely as possible, a set of samples $(\mathbf{u}^m, \mathbf{v}^m)$ for $m = 1, 2, \ldots, N_S$. Such a task does not make sense for a model that has a stochastic rather than deterministic relationship between its input and output activities. Instead, stochastic networks are appropriate for representing statistical aspects of the relationship between two sets of variables. For example, suppose that we drew sample pairs from a joint probability distribution $P[\mathbf{u}, \mathbf{v}] = P[\mathbf{v}|\mathbf{u}]P[\mathbf{u}]$. In this case, a given value of $\mathbf{u}^m$, chosen from the distribution $P[\mathbf{u}]$, is associated with another vector $\mathbf{v}^m$ stochastically rather than deterministically. The probability that a particular output $\mathbf{v}^m$ is associated with $\mathbf{u}^m$ is given by $P[\mathbf{v}^m|\mathbf{u}^m]$. In other words, $P[\mathbf{u}]$ describes the probability of various $\mathbf{u}$ vectors appearing, and $P[\mathbf{v}|\mathbf{u}]$ is the probability that a particular vector $\mathbf{u}$ is associated with another vector $\mathbf{v}$.

A natural supervised learning task for a stochastic network is to make the input-output distribution of the network, $P[\mathbf{v}|\mathbf{u}; \mathbf{W}]$, match as closely as possible, the probability distribution $P[\mathbf{v}|\mathbf{u}]$ associated with the samples $(\mathbf{u}^m, \mathbf{v}^m)$. This is done by adjusting the feedforward weight matrix $\mathbf{W}$. Note that we are using the argument $\mathbf{W}$ to distinguish between two different distributions, $P[\mathbf{u}|\mathbf{v}]$, which is provided externally and generates the sample data, and $P[\mathbf{u}|\mathbf{v}; \mathbf{W}]$, which is the distribution generated by the Boltzmann machine with weight matrix $\mathbf{W}$. The idea of constructing networks that reproduce probability distributions inferred from sample data is central to the problem of density estimation, which is covered more fully in chapter 10.

density estimation

We first consider a Boltzmann machine with only feedforward weights $\mathbf{W}$ connecting $\mathbf{u}$ to $\mathbf{v}$, and no recurrent weight among the $\mathbf{v}$ units. Given input $\mathbf{u}$, an output $\mathbf{v}$ is computed by setting each component v_a to 1 with probability $F(\sum_b W_{ab} u_b)$ (and 0 otherwise) where $F(I) = 1/(1 + \exp(-I))$. This is the Gibbs sampling procedure discussed in chapter 7 applied to the feedforward Boltzmann machine. Because there are no recurrent connections, the states of the output units are independent, and they can all be sampled simultaneously. Analogous to the discussion in chapter 7, this

4 Information Theory

4.1 Entropy and Mutual Information

Neural encoding and decoding focus on the question "What does the response of a neuron tell us about a stimulus?" In this chapter we consider a related but different question "How much does the neural response tell us about a stimulus?" The techniques of information theory allow us to answer this question in a quantitative manner. Furthermore, we can use them to ask what forms of neural response are optimal for conveying information about natural stimuli. Information theoretic principles play an important role in many of the unsupervised learning methods that are discussed in chapters 8 and 10.

Shannon invented information theory as a general framework for quantifying the ability of a coding scheme or a communication channel (such as the optic nerve) to convey information. It is assumed that the code involves a number of symbols (such as different neuronal responses), and that the coding and transmission processes are stochastic and noisy. The quantities we consider in this chapter, the entropy and the mutual information, depend on the probabilities with which these symbols, or combinations of them, are used. Entropy is a measure of the theoretical capacity of a code to convey information. Mutual information measures how much of that capacity is actually used when the code is employed to describe a particular set of data. Communication channels, if they are noisy, have only limited capacities to convey information. The techniques of information theory are used to evaluate these limits and find coding schemes that saturate them.

In neuroscience applications, the symbols we consider are neuronal responses, and the data sets they describe are stimulus characteristics. In the most complete analyses, which are considered at the end of the chapter, the neuronal response is characterized by a list of action potential firing times. The symbols being analyzed in this case are sequences of action potentials. Computing the entropy and mutual information for spike sequences can be difficult because the frequency of occurrence of many different spike sequences must be determined. This typically requires a large amount of

$$\mathbf{w} \to \mathbf{w} - \epsilon_w \nabla_{\mathbf{w}} E \quad \text{or} \quad w_b \to w_b - \epsilon_w \frac{\partial E}{\partial w_b}, \tag{8.58}$$

where $\nabla_{\mathbf{w}} E$ is the vector with components $\partial E / \partial w_b$. This rule is sensible because $-\nabla_{\mathbf{w}} E$ points in the direction (in the space of synaptic weights) along which E decreases most rapidly. This process tends to reduce E because, for small ϵ_w,

$$E(\mathbf{w} - \epsilon_w \nabla_{\mathbf{w}} E) \approx E(\mathbf{w}) - \epsilon_w |\nabla_{\mathbf{w}} E|^2 \leq E(\mathbf{w}). \tag{8.59}$$

If ϵ_w is too large or $\mathbf{w}$ is very near to a point where $\nabla_{\mathbf{w}} E(\mathbf{w}) = \mathbf{0}$, E can increase instead. We assume that ϵ_w is small enough so that E decreases at least until $\mathbf{w}$ is very close to a minimum. If E has many minima, gradient descent will lead to only one of them (a local minimum), and not necessarily the one with the lowest value of E (the global minimum). In the case of function approximation using basis functions as in equation 8.51, gradient descent finds a value of $\mathbf{w}$ that satisfies the normal equations, and therefore constructs an optimal function approximator, because the error function of equation 8.52 has only one minimum.

For function approximation, the error E in equation 8.52 is a sum over the set of examples. As a result, $\nabla_{\mathbf{w}} E$ also involves a sum,

$$\nabla_{\mathbf{w}} E = -\sum_{m=1}^{N_S} (h(s^m) - v(s^m)) \mathbf{f}(s^m), \tag{8.60}$$

where we have used the fact that $\nabla_{\mathbf{w}} v(s^m) = \mathbf{f}(s^m)$. The presence of the sum means that the learning rule of equation 8.58 cannot be applied until all the sample patterns have been presented, because all of them are needed to compute the amount by which the weight vector should be changed. It is much more convenient if updating of the weights takes place continuously while sample inputs are presented. This can be done using a procedure known as stochastic gradient descent. This alternative procedure involves presenting randomly chosen input-output pairs s^m and $h(s^m)$, and change $\mathbf{w}$ according to

stochastic gradient decent

$$\mathbf{w} \to \mathbf{w} + \epsilon_w (h(s^m) - v(s^m)) \mathbf{f}(s^m). \tag{8.61}$$

delta rule

This rule, called the delta rule, allows learning to take place one sample at a time. Use of this rule is based on the fact that summing the changes proportional to $(h(s^m) - v(s^m))\mathbf{f}(s^m)$ over the random choices of m is, on average, equivalent to doing the sum in equation 8.60. The effect of using equation 8.61 instead of equation 8.58 is the introduction of noise that causes the weights to fluctuate about average values that satisfy the normal equations. Replacing a full sum by an appropriately weighted sum of randomly chosen terms is an example of a so-called Monte Carlo method. There are more efficient methods of searching for minima of functions than stochastic gradient descent, but many of them are complicated to implement.

Figure 8.14 shows the result of modifying an initially random set of weights using the delta rule. Ultimately, an array of input neurons with

The minus sign makes h a decreasing function of its argument, as required. Note that information is really a dimensionless number. The bit, like the radian for angles, is not a dimensional unit but a reminder that a particular system is being used.

Expression (4.2) quantifies the surprise or unpredictability associated with a particular response. Shannon's entropy is just this measure averaged over all responses,

entropy

$$H = -\sum_r P[r] \log_2 P[r] \,. \tag{4.3}$$

In the sum that determines the entropy, the factor $h = -\log_2 P[r]$ is multiplied by the probability that the response with rate r occurs. Responses with extremely low probabilities may contribute little to the total entropy, despite having large h values, because they occur so rarely. In the limit when $P[r] \to 0$, $h \to \infty$, but an event that does not occur does not contribute to the entropy because the problematic expression $-0 \log_2 0$ is evaluated as $-\epsilon \log_2 \epsilon$ in the limit $\epsilon \to 0$, which is 0. Very high probability responses also contribute little because they have $h \approx 0$. The responses that contribute most to the entropy have high enough probabilities so that they appear with a fair frequency, but not high enough to make h too small.

Computing the entropy in some simple cases helps provide a feel for what it measures. First, imagine the least interesting situation: when a neuron responds every time by firing at the same rate. In this case, all of the probabilities $P[r]$ are 0, except for one of them, which is 1. This means that every term in the sum of equation (4.3) is 0 because either $P[r] = 0$ or $\log_2 1 = 0$. Thus, a set of identical responses has zero entropy. Next, imagine that the neuron responds in only two possible ways, either with rate r_+ or r_-. In this case, there are only two nonzero terms in equation (4.3), and, using the fact that $P[r_-] = 1 - P[r_+]$, the entropy is

$$H = -(1 - P[r_+]) \log_2 (1 - P[r_+]) - P[r_+] \log_2 P[r_+] \,. \tag{4.4}$$

This entropy, plotted in figure 4.1A, takes its maximum value of 1 bit when $P[r_-] = P[r_+] = 1/2$. Thus, a code consisting of two equally likely responses has one bit of entropy.

Mutual Information

To convey information about a set of stimuli, neural responses must be different for different stimuli. Entropy is a measure of response variability, but it does not tell us anything about the source of that variability. A neuron can provide information about a stimulus only if its response variability is correlated with changes in that stimulus, rather than being purely random or correlated with other unrelated factors. One way to determine whether response variability is correlated with stimulus variability is to compare the responses obtained using a different stimulus on every trial with those measured in trials involving repeated presentations of the same

vided with a sequence of N_S sample stimuli, s^m for $m = 1, 2, \ldots, N_S$, and the corresponding function values $h(s^m)$, during a training period. To make $v(s^m)$ match $h(s^m)$ as closely as possible for all m, we minimize the error

$$E = \frac{1}{2}\sum_{m=1}^{N_S}\left(h(s^m) - v(s^m)\right)^2 = \frac{N_S}{2}\left\langle (h(s) - \mathbf{w}\cdot\mathbf{f}(s))^2\right\rangle . \tag{8.52}$$

We have made the replacement $v(s) = \mathbf{w}\cdot\mathbf{f}(s)$ in this equation and have used the bracket notation for the average over the training samples. Equations for the weights that minimize this error, called the normal equations, are obtained by setting its derivative with respect to the weights to 0, yielding the condition

normal equations

$$\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \mathbf{w} = \langle \mathbf{f}(s)h(s)\rangle . \tag{8.53}$$

The supervised Hebbian rule of equation 8.45, applied in this case, ultimately sets the weight vector to $\mathbf{w} = \langle \mathbf{f}(s)h(s)\rangle/\alpha$. These weights must satisfy the normal equations 8.53 if they are to optimize function approximation. There are two circumstances under which this occurs. The obvious one is when the input units are orthogonal across the training stimuli, $\langle \mathbf{f}(s)\mathbf{f}(s)\rangle = \mathbf{I}$. In this case, the normal equations are satisfied with $\alpha = 1$. However, this condition is unlikely to hold for most sets of input tuning curves. An alternative possibility is that for all pairs of stimuli s^m and s^n in the training set,

$$\mathbf{f}(s^m)\cdot\mathbf{f}(s^n) = c\delta_{mn} \tag{8.54}$$

for some constant c. This is called a tight frame condition. If it is satisfied, the weights given by supervised Hebbian learning with decay can satisfy the normal equations. To see this, we insert the weights $\mathbf{w} = \langle \mathbf{f}(s)h(s)\rangle/\alpha$ into equation 8.53 and use 8.54 to obtain

tight frame

$$\begin{aligned}\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \mathbf{w} &= \frac{\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \langle \mathbf{f}(s)h(s)\rangle}{\alpha} = \frac{1}{\alpha N_S^2}\sum_{mn}\mathbf{f}(s^m)\mathbf{f}(s^m)\cdot\mathbf{f}(s^n)h(s^n)\\ &= \frac{c}{\alpha N_S^2}\sum_m \mathbf{f}(s^m)h(s^m) = \frac{c}{\alpha N_S}\langle \mathbf{f}(s)h(s)\rangle .\end{aligned} \tag{8.55}$$

This shows that the normal equations are satisfied for $\alpha = c/N_S$. Thus, we have shown two ways that supervised Hebbian learning can solve the function approximation problem, but both require special conditions on the basis functions $\mathbf{f}(s)$. A more general scheme, discussed below, involves using an error-correcting rule.

Supervised Error-Correcting Rules

An essential limitation of supervised Hebbian rules is that synaptic modification does not depend on the actual performance of the network. An

full response entropy, which from equations 4.3 and 4.6 gives

$$I_{\mathrm{m}} = H - H_{\mathrm{noise}} = -\sum_r P[r]\log_2 P[r] + \sum_{s,r} P[s]P[r|s]\log_2 P[r|s]\,. \quad (4.7)$$

The probability of a response r is related to the conditional probability $P[r|s]$ and the probability $P[s]$ that stimulus s is presented by the identity (chapter 3),

$$P[r] = \sum_s P[s]P[r|s]\,. \quad (4.8)$$

Using this, and writing the difference of the two logarithms in equation 4.7 as the logarithm of the ratio of their arguments, we can rewrite the mutual information as

mutual information

$$I_{\mathrm{m}} = \sum_{s,r} P[s]P[r|s]\log_2\left(\frac{P[r|s]}{P[r]}\right)\,. \quad (4.9)$$

Recall from chapter 3 that

$$P[r,s] = P[s]P[r|s] = P[r]P[s|r]\,, \quad (4.10)$$

where $P[r,s]$ is the joint probability of stimulus s appearing and response r being evoked. Equation 4.10 can be used to derive yet another form for the mutual information,

$$I_{\mathrm{m}} = \sum_{s,r} P[r,s]\log_2\left(\frac{P[r,s]}{P[r]P[s]}\right)\,. \quad (4.11)$$

This equation reveals that the mutual information is symmetric with respect to interchange of s and r, which means that the mutual information that a set of responses conveys about a set of stimuli is identical to the mutual information that the set of stimuli conveys about the responses. To see this explicitly, we apply equation 4.10 again to write

$$I_{\mathrm{m}} = -\sum_s P[s]\log_2 P[s] + \sum_{s,r} P[r]P[s|r]\log_2 P[s|r]\,. \quad (4.12)$$

This result is the same as equation 4.7, except that the roles of the stimulus and the response have been interchanged. Equation 4.12 shows how response variability limits the ability of a spike train to carry information. The second term on the right side, which is negative, is the average uncertainty about the identity of the stimulus given the response, and reduces the total stimulus entropy represented by the first term.

To provide some concrete examples, we compute the mutual information for a few simple cases. First, suppose that the responses of the neuron are completely unaffected by the identity of the stimulus. In this case, $P[r|s] = P[r]$, and from equation 4.9 it follows immediately that $I_{\mathrm{m}} = 0$. At the other extreme, suppose that each stimulus s produces a unique and

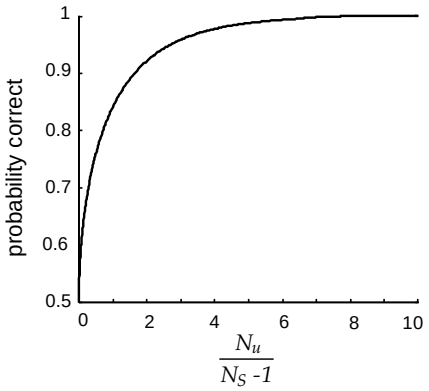

Figure 8.12 Percentage of correct responses for a perceptron with a Hebbian weight vector for a random binary input-output map. As the ratio of the number of inputs, N_u, to one less than the number of input vectors being learned, $N_S - 1$, grows, the percentage of correct responses goes to 1. When this ratio is small, the percentage of correct responses approaches the chance level of $1/2$.

are chosen randomly with equal probabilities of being $+1$ or -1. Including the dot product, the right side of the expression $N_u \eta^n = \sum_{m \neq n} v^m \mathbf{u}^m \cdot \mathbf{u}^n$ defining η^n is the sum of $(N_S - 1)N_u$ terms, each of which is equally likely to be $+1$ or -1. For large N_u and N_S, the central limit theorem (see the Mathematical Appendix) implies that the distribution of η values is Gaussian with mean 0 and variance $(N_S - 1)/N_u$. This suggests that the perceptron with Hebbian weights should work well if the number of input patterns being learned is significantly less than the number of input vector components. We can make this more precise by noting from equations 8.46 with $\gamma = 0$ and equation 8.49 that, for $v^n = +1$, the perceptron will give the correct answer if $-1 < \eta^n < \infty$. Similarly, for $v^n = -1$, the perceptron will give the correct answer if $-\infty < \eta^n < 1$. If v^n has probability $1/2$ of taking either value, the probability of the perceptron giving the correct answer is $1/2$ times the integral of the Gaussian distribution from -1 to ∞ plus $1/2$ times its integral from $-\infty$ to 1. Combining these two terms, we find

$$P[\text{correct}] = \sqrt{\frac{N_u}{2\pi(N_S - 1)}} \int_{-\infty}^{1} d\eta \, \exp\left(-\frac{N_u \eta^2}{2(N_S - 1)}\right). \tag{8.50}$$

This result is plotted in figure 8.12, which shows that the Hebbian perceptron performs quite well if $N_S - 1$ is less than about $0.2N_u$. It is possible for the perceptron to perform considerably better than this if a non-Hebbian weight vector is used. We return to this in a later section.

Function Approximation

In chapter 1, we studied examples in which the firing rate of a neuron was given by a function of a stimulus parameter, namely, the response tuning curve. When such a relationship exists, we can think of the neuronal

interchange of P and Q. Comparing the definition 4.15 with equation 4.11, we see that the mutual information is the KL divergence between the distributions $P[r, s]$ and $P[r]P[s]$. If the stimulus and the response were independent of one another, $P[r, s]$ would be equal to $P[r]P[s]$. Thus, the mutual information is the KL divergence between the actual probability distribution $P[r, s]$ and the value it would take if the stimulus and response were independent. The fact that $D_{\rm KL} \geq 0$ proves that the mutual information cannot be negative. In addition, it can never be larger than either the full response entropy or the entropy of the stimulus set.

Entropy and Mutual Information for Continuous Variables

Up to now we have characterized neural responses using discrete spike-count rates. As in chapter 3, it is often convenient to treat these rates instead as continuous variables. There is a complication associated with entropies that are defined in terms of continuous response variables. If we could measure the value of a continuously defined firing rate with unlimited accuracy, it would be possible to convey an infinite amount of information using the endless sequence of decimal digits of this single variable. Of course, practical considerations always limit the accuracy with which a firing rate can be measured or conveyed.

To define the entropy associated with a continuous measure of a neural response, we must include some limit on the measurement accuracy. The effects of this limit typically cancel in computations of mutual information because the mutual information is the difference between two entropies. In this section, we show how entropy and mutual information are computed for responses characterized by continuous firing rates. For completeness, we also treat the stimulus parameter s as a continuous variable. This means that the probability $P[s]$ is replaced by the probability density $p[s]$, and sums over s are replaced by integrals.

For a continuously defined firing rate, the probability of the firing rate lying in the range between r and $r + \Delta r$, for small Δr, is expressed in terms of a probability density as $p[r]\Delta r$. Summing over discrete bins of size Δr, we find, by analogy with equation (4.3),

$$\begin{aligned} H &= -\sum p[r]\Delta r \log_2(p[r]\Delta r) \\ &= -\sum p[r]\Delta r \log_2 p[r] - \log_2 \Delta r \,. \end{aligned} \tag{4.16}$$

To extract the last term we have expressed the logarithm of a product as the sum of two logarithms and used the fact that the sum of the response probabilities is 1. We would now like to take the limit $\Delta r \to 0$ but we cannot, because the $\log_2 \Delta r$ term diverges in this limit. This divergence reflects the fact that a continuous variable measured with perfect accuracy has infinite entropy. However, for reasonable (i.e., Riemann integrable) $p[r]$, everything works out fine for the first term because the sum becomes

the superscript is a label and does not signify either a component of $\mathbf{u}$ or a power of v. For a feedforward network, an averaged Hebbian plasticity rule for supervised learning can be obtained from equation 8.4 by averaging across all the input-output pairs,

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle = \frac{1}{N_S} \sum_{m=1}^{N_S} v^m \mathbf{u}^m \,. \tag{8.44}$$

cross-correlation

As in the unsupervised Hebbian learning case, the synaptic modification process depends on the input-output cross-correlation $\langle v\mathbf{u} \rangle$. However, for supervised learning, the output $v = v^m$ is imposed on the network rather than being determined by it.

Unless the cross-correlation is 0, equation 8.44 never stops changing the synaptic weights. The methods introduced to stabilize Hebbian modification in the case of unsupervised learning can be applied to supervised learning as well. However, stabilization is easier in the supervised case, because the right side of equation 8.44 does not depend on $\mathbf{w}$. Therefore, the growth is only linear rather than exponential in time, making a simple multiplicative synaptic weight decay term sufficient for stability. This is introduced by writing the supervised learning rule as

supervised learning with decay

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle - \alpha \mathbf{w} \,, \tag{8.45}$$

for some positive constant α. Asymptotically, equation 8.45 makes $\mathbf{w} = \langle v\mathbf{u} \rangle / \alpha$, that is, the weights become proportional to the input-output cross-correlation.

We discuss supervised Hebbian learning in the case of a single output unit, but the results can be generalized to multiple outputs.

Classification and the Perceptron

perceptron
binary classifier

The perceptron is a nonlinear map that classifies inputs into one of two categories. It thus acts as a binary classifier. To make the model consistent when units are connected in a network, we also require the inputs to be binary. We can think of the two possible states as representing units that are either active or inactive. As such, we would naturally assign them the values 1 and 0. However, the analysis is simpler (while producing similar results) if, instead, we require the inputs u_a and output v to take the two values $+1$ and -1.

The output of the perceptron is based on a modification of the linear rule of equation 8.2 to

$$v = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{u} - \gamma \geq 0 \\ -1 & \text{if } \mathbf{w} \cdot \mathbf{u} - \gamma < 0 \,. \end{cases} \tag{8.46}$$

implications of an observed response selectivity. For example, we might ask whether neural responses to natural stimuli are optimized to convey as much information as possible. This hypothesis can be tested by computing the response characteristics that maximize the mutual information conveyed about naturally occurring stimuli and comparing the results with responses observed experimentally.

Because the mutual information is the full response entropy minus the noise entropy, maximizing the information involves a compromise. We must make the response entropy as large as possible without allowing the noise entropy to get too big. If the noise entropy is small, maximizing the response entropy, subject to an appropriate constraint, maximizes the mutual information to a good approximation. We therefore begin our discussion by studying how response entropy can be maximized. Later in the discussion, we will consider the effects of noise entropy.

Constraints play a crucial role in this analysis. We have already seen that the theoretical information-carrying capacity associated with a continuous firing rate is limited only by the resolution with which the firing rate can be defined. Even with a finite resolution, a firing rate could convey an infinite amount of information if it could take arbitrarily high values. Thus, we must impose some constraint that limits the firing rate to a realistic range. Possible constraints include limiting the maximum allowed firing rate or holding the average firing rate or its variance fixed.

Entropy Maximization for a Single Neuron

To maximize the response entropy, we must find a probability density $p[r]$ that makes the integral in equation 4.17 as large as possible while satisfying whatever constraints we impose. During the maximization process, the resolution Δr is held fixed, so the $\log_2 \Delta r$ term remains constant, and it can be ignored. As a result, it will not generally appear in the following equations. One constraint that always applies in entropy maximization is that the integral of the probability density must be 1. Suppose that the neuron in question has a maximum firing rate of $r_{\max}$. Then, the integrals in question extend from 0 to $r_{\max}$. To find the $p[r]$ producing the maximum entropy, we must maximize

$$-\int_0^{r_{\max}} dr\, p[r] \log_2 p[r]\,, \tag{4.20}$$

subject to the constraint

$$\int_0^{r_{\max}} dr\, p[r] = 1\,. \tag{4.21}$$

The result, computed using Lagrange multipliers (see the Mathematical Appendix), is that the probability density that maximizes the entropy sub-

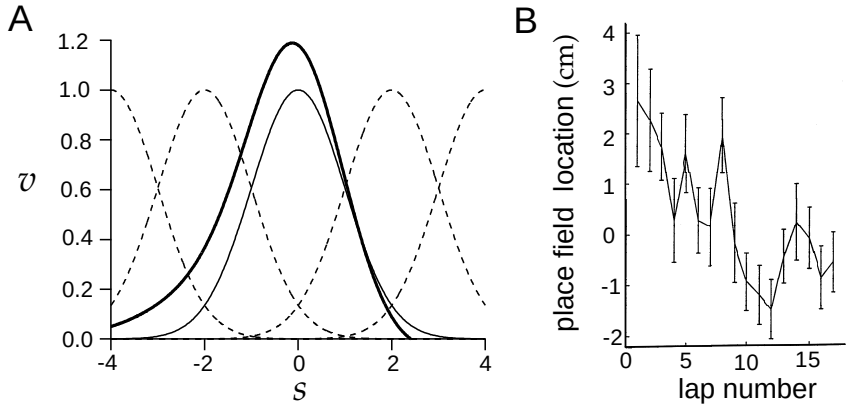

Figure 8.11 Predicted and experimental shifts of place fields. (A) Shift in a neuronal firing-rate tuning curve caused by repeated exposure to a time-dependent stimulus during training. The dashed curves and thin solid curve indicate the initial response tuning curves of a network of interconnected neurons. The thick solid curve is the response tuning curve of the neuron that initially had the thin solid tuning curve after a training period involving a time-dependent stimulus. The tuning curve increases in amplitude, asymmetrically broadens, and shifts as a result of temporally asymmetric Hebbian plasticity. The shift shown corresponds to a training stimulus with a positive rate of change, that is, one that moves rightward on this plot as a function of time. The corresponding shift in the tuning curve is to the left. The shift has been calculated using more neurons and tuning curves than are shown in this plot. (B) Location of place field centers while a rat traversed laps around a closed track (0 is defined as the average center location across the whole experiment). Over sequential laps, the place fields shifted backward relative to the direction the rat moved. (A adapted and modified from Abbott & Blum, 1996; B adapted from Mehta et al., 1997.)

which reaches a maximum value for the optimal stimulus $s = s_a$. Different neurons have different optimal stimulus values, as depicted by the dashed and thin solid curves in figure 8.11A. We now examine what happens when the plasticity rule 8.18 is applied throughout a training period during which the stimulus being presented is an increasing function of time, i.e., moves to the right in figure 8.11A. Such a stimulus excites the different neurons in the network sequentially. For example, the neuron with $s_a = -2$ is active before the neuron with $s_a = 0$, which in turn is active before the neuron with $s_a = 2$. If the stimulus changes rapidly enough, the interval between the firing of the neuron with $s_a = -2$ and that with $s_a = 0$ will fall within the window for LTP depicted in figure 8.2B. This means that a synapse from the neuron with $s_a = -2$ to the $s_a = 0$ neuron will be strengthened. On the other hand, because the neuron with $s_a = 2$ fires after the $s_a = 0$ neuron, falling within the window for LTD, a synapse from it to the $s_a = 0$ neuron will be weakened.

The effect of this type of modification on the tuning curve in the middle of the array (the thin solid curve in figure 8.11A centered at $s = 0$) is shown by the thick solid curve in figure 8.11A. After the training period, the neuron with $s_a = 0$ receives strengthened input from neurons with $s_a < 0$ and

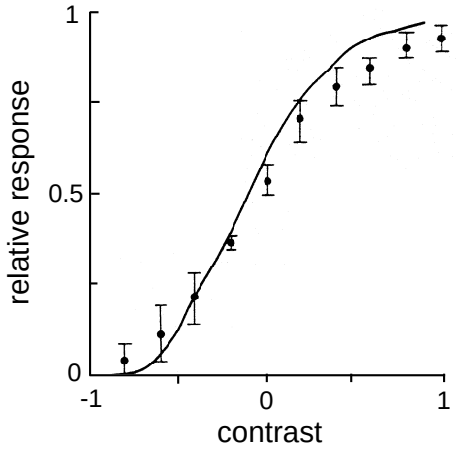

Figure 4.2 Contrast response of the fly LMC (data points) compared to the integral of the natural contrast probability distribution (solid curve). The relative response is the amplitude of the membrane potential fluctuation produced by the onset of a light or dark image with a given level of contrast divided by the maximum response. Contrast is defined relative to the background level of illumination. (Adapted from Laughlin, 1981.)

Even though neurons have maximum firing rates, the constraint $r \leq r_{\max}$ may not always be the factor limiting the entropy. For example, the average firing rate of the neuron may be constrained to values much less than $r_{\max}$, or the variance of the firing rate might be constrained. The reader is invited to show that the entropy-maximizing probability density, if the average firing rate is constrained to a fixed value, is an exponential. A related calculation shows that the probability density that maximizes the entropy subject to constraints on the firing rate and its variance is a Gaussian.

Populations of Neurons

When a population of neurons encodes a stimulus, optimizing their individual response properties will not necessarily lead to an optimized population response. Optimizing individual responses could result in a highly redundant population representation in which different neurons encode the same information. Entropy maximization for a population requires that the neurons convey independent pieces of information (i.e., they must have different response selectivities). Let the vector $\mathbf{r}$ with components r_a for $a = 1, 2, \ldots, N$ denote the firing rates for a population of N neurons, measured with resolution Δr. If $p[\mathbf{r}]$ is the probability of evoking a population response characterized by the vector $\mathbf{r}$, the entropy for the entire population response is

$$H = -\int d\mathbf{r}\, p[\mathbf{r}] \log_2 p[\mathbf{r}] - N \log_2 \Delta r\,. \tag{4.26}$$

Along with the full population entropy of Equation 4.26, we can also con-

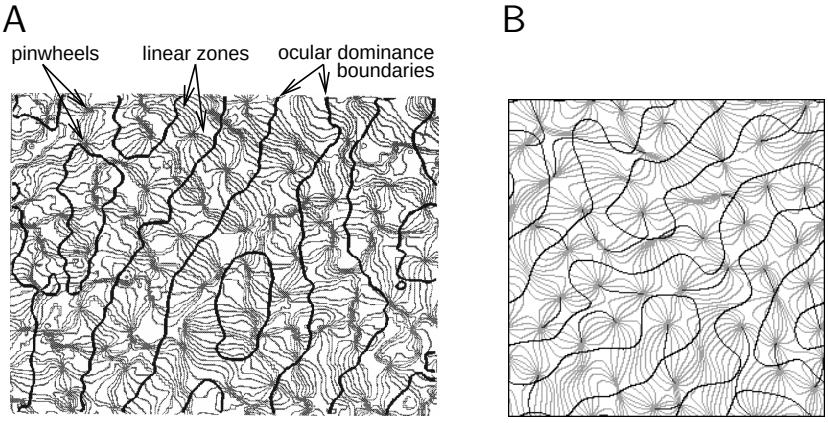

Figure 8.10 Orientation domains and ocular dominance. (A) Contour map showing iso-orientation contours (gray lines) and the boundaries of ocular dominance stripes (black lines) in a 1.7 × 1.7 mm patch of macaque primary visual cortex. Iso-orientation contours are drawn at intervals of 11.25°. Pinwheels are singularities in the orientation map where all the orientations meet, and linear zones are extended patches over which the iso-orientation contours are parallel. (B) Ocular dominance and orientation map produced by the elastic net model. The significance of the lines is the same as in (A), except that the darker gray lines show orientation preferences of 0°. (A adapted from Obermayer & Blasdel, 1993; B from Erwin et al., 1995.)

provides a good illustration of this problem. In the absence of recurrent connections, this rule sets each row of the feedforward weight matrix to the principal eigenvector of the input correlation matrix, making each output unit respond identically.

One way to reduce redundancy in a linear model is to make the linear recurrent interactions of equation 8.29 plastic rather than fixed, using an anti-Hebbian modification rule. As the name implies, anti-Hebbian plasticity causes synapses to decrease (rather than increase) in strength when there is simultaneous pre- and postsynaptic activity. The recurrent interactions arising from an anti-Hebb rule can prevent different output units from representing the same eigenvector. This occurs because the recurrent interactions tend to make output units less correlated by canceling the effects of common feedforward input. Anti-Hebbian modification is believed to be the predominant form of plasticity at synapses from parallel fibers to Purkinje cells in the cerebellum, although this may be a special case because Purkinje cells inhibit rather than excite their targets. A basic anti-Hebb rule for $M_{aa'}$ can be created simply by changing the sign on the right side of equation 8.3. However, just as Hebbian plasticity tends to make weights increase without bound, anti-Hebbian modification tends to make them decrease to 0, and to avoid this it is necessary to use

$$\tau_M \frac{d\mathbf{M}}{dt} = -\mathbf{v}\mathbf{v} + \beta\mathbf{M} \quad \text{or} \quad \tau_M \frac{dM_{aa'}}{dt} = -v_a v_{a'} + \beta M_{aa'} \tag{8.40}$$

to modify the off-diagonal components of $\mathbf{M}$ (the diagonal components are defined to be 0). Here, β is a positive constant. For suitably chosen β and τ_M, the combination of rules 8.39 and 8.40 produces a stable configuration

Exact factorization and probability equalization are difficult to achieve, especially if the form of the neural response is restricted. These goals are likely to be impossible to achieve, for example, if the neural responses are modeled as having a linear relation to the stimulus. A more modest goal is to require that the lowest-order moments of the population-response probability distribution match those of a fully factorized and equalized distribution. If the individual response probability distributions are equal, the average firing rates and firing rate variances will be the same for all neurons, $\langle r_a \rangle = \langle r \rangle$ and $\langle (r_a - \langle r \rangle)^2 \rangle = \sigma_r^2$ for all a. Furthermore, the covariance matrix for a factorized and probability-equalized population distribution is proportional to the identity matrix,

$$Q_{ab} = \int d\mathbf{r}\, p[\mathbf{r}](r_a - \langle r \rangle)(r_b - \langle r \rangle) = \sigma_r^2 \delta_{ab} \,. \tag{4.31}$$

decorrelation and variance equalization

Finding response distributions that satisfy only the decorrelation and variance equalization condition of equation 4.31 is usually tractable. In the following examples, we restrict ourselves to this easier task. This maximizes the entropy only if the statistics of the responses are Gaussian, but it is a reasonable procedure even in a non-Gaussian case, because it typically reduces the redundancy in the population code and spreads the load of information transmission equally among the neurons.

Application to Retinal Ganglion Cell Receptive Fields

Entropy and information maximization have been used to explain properties of visual receptive fields in the retina, LGN, and primary visual cortex. The basic assumption is that these receptive fields serve to maximize the amount of information that the associated neural responses convey about natural visual scenes in the presence of noise. Information theoretical analyses are sensitive to the statistical properties of the stimuli being represented, so the statistics of natural scenes play an important role in these studies. Natural scenes exhibit substantial spatial and temporal redundancy. Maximizing the information conveyed requires removing this redundancy from the neural responses.

It should be kept in mind that the information maximization approach sets limited goals and requires strong assumptions about the nature of the constraints relevant to the nervous system. In addition, the approach analyzes only the representational properties of neural responses and ignores the computational goals of the visual system, such as object recognition or target tracking. Finally, maximizing other measures of performance, different from the mutual information, may give similar results. Nevertheless, the principle of information maximization is quite successful at accounting for properties of receptive fields early in the visual pathway.

In chapter 2, a visual image was defined by a contrast function $s(x, y, t)$ with a trial-averaged value of 0. For the calculations we present here, it is more convenient to express the x and y coordinates for locations on the

receptive field) and the preferred eye for neuron a. Map development in such a model is studied by noting how the appearance of various stimulus features and neuronal selectivities affects the matrix $\mathbf{W}$. By associating the index a with cortical location, the structure of the final matrix $\mathbf{W}$ that arises from a plasticity rule predicts how selectivities are mapped across the cortex.

The activity of a particular output unit in a feature-based model is determined by how closely the stimulus being presented matches its preferred stimulus. The weights W_{ab} for all b values determine the preferred stimulus features for neuron a, and we assume that the activation of neuron a is high if the components of the input u_b match the components of W_{ab}. A convenient way to achieve this is to express the activation for unit a as $\exp(-\sum_b (u_b - W_{ab})^2/(2\sigma_b^2))$, which has its maximum at $u_b = W_{ab}$ for all b, and falls off as a Gaussian function for less perfect matches of the stimulus to the selectivity of the cell. The parameter σ_b determines how selective the neuron is for characteristic b of the stimulus.

The Gaussian expression for the activation of neuron a is not used directly to determine its level of activity. Rather, as in the case of competitive Hebbian learning, we introduce a competitive activity variable for cortical site a,

$$z_a = \frac{\exp\left(-\sum_b (u_b - W_{ab})^2/(2\sigma_b^2)\right)}{\sum_{a'} \exp\left(-\sum_b (u_b - W_{a'b})^2/(2\sigma_b^2)\right)}. \tag{8.36}$$

In addition, some cooperative mechanism must be included to keep the maps smooth, which means that nearby neurons should, as far as possible, have similar selectivities. The two algorithms we discuss, the self-organizing map and the elastic net, differ in how they introduce this second element.

self-organizing map

The self-organizing map spreads the activity defined by equation 8.36 to nearby cortical sites through equation 8.35, $v_a = \sum_{a'} M_{aa'} z_{a'}$. This gives cortical cells a and a' similar selectivities if they are nearby, because v_a and $v_{a'}$ affect one another through local recurrent excitation. The elastic net sets the activity of unit a to the result of equation 8.36, $v_a = z_a$, which generates competition. Smoothness of the map is ensured not by spreading this activity, as in the self-organizing map, but by including an additional term in the plasticity rule that tends to make nearby selectivities the same (see below).

elastic net

Hebbian development of the selectivities characterized by $\mathbf{W}$ is generated by an activity-dependent rule. In general, Hebbian plasticity adjusts the weights of activated units so that they become more responsive to, and selective for, input patterns that excite them. Feature-based models achieve the same thing by modifying the selectivities W_{ab} so they more closely match the input parameters u_b when output unit a is activated by $\mathbf{u}$. For the case of the self-organized map, this is achieved through the averaged

feature-based learning rule

set of $\vec{a}$ values with a vector $\vec{a}$ that is allowed to vary continuously. In other words, as an approximation, we proceed as if there were a neuron corresponding to every continuous value of $\vec{a}$. This allows us to treat $L(\vec{a})$ as a function of $\vec{a}$ and to replace sums over neurons with integrals over $\vec{a}$. In the case we are considering, the receptive fields of retinal ganglion cells cover the retina densely, with many receptive fields overlapping each point on the retina, so the replacement of discrete sums over neurons with continuous integrals over $\vec{a}$ is quite accurate.

The Whitening Filter

We will not attempt a complete entropy maximization for the case of retinal ganglion cells. Instead, we follow the approximate procedure of setting the correlation matrix between different neurons within the population proportional to the identity matrix (equation 4.31). The relevant correlation is the average, over all stimuli, of the product of the linear responses of two cells, with receptive fields centered at $\vec{a}$ and $\vec{b}$,

$$Q_{\rm LL}(\vec{a},\vec{b}) = \langle L_{\rm s}(\vec{a})L_{\rm s}(\vec{b})\rangle = \int d\vec{x}\, d\vec{y}\, D_{\rm s}(\vec{x}-\vec{a})D_{\rm s}(\vec{y}-\vec{b})\langle s_{\rm s}(\vec{x})s_{\rm s}(\vec{y})\rangle\,. \tag{4.36}$$

The average here, denoted by angle brackets, is not over trials but over the set of natural scenes for which we believe the receptive field is optimized. By analogy with equation 4.31, decorrelation and variance equalization of the different retinal ganglion cells, when $\vec{a}$ and $\vec{b}$ are taken to be continuous variables, require that we set this correlation function proportional to a δ function,

$$Q_{\rm LL}(\vec{a},\vec{b}) = \sigma_{\rm L}^2\delta(\vec{a}-\vec{b})\,. \tag{4.37}$$

This is the continuous variable analog of making a discrete correlation matrix proportional to the identity matrix (equation 4.31). The δ function with vector arguments is nonzero only when all of the components of $\vec{a}$ and $\vec{b}$ are identical.

The quantity $\langle s_{\rm s}(\vec{x})s_{\rm s}(\vec{y})\rangle$ in equation 4.36 is the correlation function of the stimulus averaged over natural scenes. Our assumption of homogeneity implies that this quantity is only a function of the vector difference $\vec{x}-\vec{y}$ (actually, if all directions are equivalent, it is only a function of the magnitude $|\vec{x}-\vec{y}|$), and we write it as

$$Q_{ss}(\vec{x}-\vec{y}) = \langle s_{\rm s}(\vec{x})s_{\rm s}(\vec{y})\rangle\,. \tag{4.38}$$

To determine the form of the receptive field filter that is optimal, we must solve equation 4.37 for D_s. This is done by expressing $D_{\rm s}$ and Q_{ss} in terms of their Fourier transforms $\tilde{D}_{\rm s}$ and $\tilde{Q}_{ss}$,

$$D_{\rm s}(\vec{x}-\vec{a}) = \frac{1}{4\pi^2}\int d\vec{\kappa}\,\exp\left(-i\vec{\kappa}\cdot(\vec{x}-\vec{a})\right)\tilde{D}_{\rm s}(\vec{\kappa}) \tag{4.39}$$

$$Q_{ss}(\vec{x}-\vec{y}) = \frac{1}{4\pi^2}\int d\vec{\kappa}\,\exp\left(-i\vec{\kappa}\cdot(\vec{x}-\vec{y})\right)\tilde{Q}_{ss}(\vec{\kappa})\,. \tag{4.40}$$

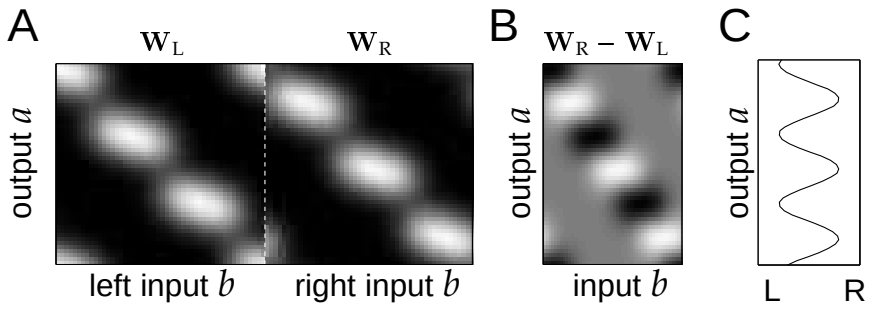

Figure 8.9 Ocular dominance patterns from a competitive Hebb rule. (A) Final stable weights W_{ab} plotted as a function of a and b, showing the relative strengths of the connections from left- and right-eye inputs centered at various retinal locations. White represents a large positive value and black represents 0. (B) The difference in the connections between right- and left-eye inputs. White indicates a positive value and black, a negative value. (C) Difference in the connections summed across all the inputs b to each cortical cell a, showing the net ocularity for each cell. The model used here has 100 input units for each eye and for the output layer, and a coarse initial topography was assumed. Circular (toroidal) boundary conditions were imposed to avoid edge effects. The input activity patterns during training represented single Gaussian illuminations in both eyes centered on a randomly chosen input unit b, with a larger magnitude for one eye (chosen randomly) than for the other. The recurrent weights $\mathbf{M}$ took the form of a Gaussian.

In the cooperative stage, the local excitation of equation 8.34 is distributed across the cortex by recurrent connections, producing a level of activity in unit a given by

$$v_a = \sum_{a'} M_{aa'} z_{a'} \, . \tag{8.35}$$

This ensures that the excitation characterized by z_a is spread across a local neighborhood of the cortex rather than being concentrated entirely at location a. In this scheme, the recurrent connections described by the matrix $\mathbf{M}$ are usually purely excitatory and of fairly short range, because the effect of longer-range inhibition has been modeled by the competition.

competitive Hebbian learning

Using the outputs of equation 8.35 in conjunction with a Hebbian rule for the feedforward weights is called competitive Hebbian learning. The competition between neurons implemented by this scheme does not ensure competition among the synapses onto a given neuron, so some mechanism such as a normalization constraint is still required. For these models, the outcome of training cannot be analyzed simply by considering eigenvectors of the covariance or correlation matrix because the activation process is nonlinear. Rather, higher-order statistics of the input distribution are important. Nonlinear competition can lead to strong differentiation of output units.

An example of the use of competitive Hebbian learning is shown in figure 8.9, in the form of a one-dimensional cortical map of ocular dominance with inputs arising from LGN neurons with receptive fields covering an extended region of the visual field (rather than the single location

a good strategy to use in an unrestricted way for visual processing. Real inputs to retinal ganglion cells involve a mixture of true signal and noise coming from biophysical sources in the retina. At high spatial frequencies, for which the true signal is weak, inputs to retinal ganglion cells are likely to be dominated by noise, especially in low-light conditions. Boosting the amplitude of this noise-dominated input and transmitting it to the brain is not an efficient visual encoding strategy.

The problem of excessive boosting of responses at high spatial frequency arises in the entropy maximization calculation because no distinction has been made between the entropy coming from true signals and that coming from noise. To correct this problem, we should maximize the information transmitted by the retinal ganglion cells about natural scenes, rather than maximize the entropy. A full information-maximization calculation of the receptive field properties of retinal ganglion cells can be performed, but this requires introducing a number of assumptions about the constraints that are relevant, and it is not entirely obvious what these constraints should be. Instead, we will follow an approximate procedure that prefilters the input to eliminate as much noise as possible, and then uses the results of this section to maximize the entropy of a linear filter acting on the prefiltered input signal.

Filtering Input Noise

Suppose that the visual stimulus on the retina is the sum of the true stimulus $s_s(\vec{x})$ that should be conveyed to the brain and a noise term $\eta(\vec{x})$ that reflects image distortion, photoreceptor noise, and other signals that are not worth conveying beyond the retina. To deal with such a mixed input signal, we express the Fourier transform of the linear kernel $\tilde{D}_s(\vec{\kappa})$ as a product of two terms: a noise filter, $\tilde{D}_\eta(\vec{\kappa})$, that eliminates as much of the noise as possible; and a whitening filter, $\tilde{D}_w(\vec{\kappa})$, that satisfies equation 4.42. The Fourier transform of the complete filter is then $\tilde{D}_s(\vec{\kappa}) = \tilde{D}_w(\vec{\kappa})\tilde{D}_\eta(\vec{\kappa})$.

To determine the form of the noise filter, we demand that when it is applied to the total input $s_s(\vec{x}) + \eta(\vec{x})$, the result is as close to the signal part of the input, $s_s(\vec{x})$, as possible. The problem of minimizing the average squared difference between the filtered noisy signal and the true signal is formally the same as the problems we solved in chapter 2 (appendix A) and chapter 3 (appendix C) to determine the optimal kernels for rate prediction and for spike decoding (also see the Mathematical Appendix). The general solution is that the Fourier transform of the optimal filter is the Fourier transform of the cross-correlation between the quantity being filtered and the quantity being approximated divided by the Fourier transform of the autocorrelation of the quantity being filtered. In the present example, there is a slight complication that the integral in equation 4.35 is not in the form of a convolution, because D_s is written as a function of $\vec{x} - \vec{a}$ rather than $\vec{a} - \vec{x}$. However, in the case we consider, this ultimately makes no difference to the final answer.

dominated by the principal eigenvector of **K**. The sign of the component $[\mathbf{w}_-]_a$ determines whether the neuron in the region of the cortex corresponding to the label a is dominated by the right eye (if it is positive) or the left eye (if it is negative). Oscillations in the signs of the components of $\mathbf{w}_-$ translate into ocular dominance stripes.

We assume that the connections between the output neurons are translation invariant, so that $\mathbf{K}_{aa'} = K(|a - a'|)$ depends only on the separation between the cortical cells a and a'. Note that it is more convenient to consider the form of the function K than to discuss the connections between output neurons (**M**) directly. We use a convenient trick to remove edge effects, which is to impose periodic boundary conditions that require the activities of the units with $a = 0$ and $a = N_v$ to be identical. This provides a reasonable model of a patch of the cortex away from regional boundaries. In some cases, edge effects can impose important constraints on the overall structure of maps, but we do not analyze this here.

In the case of periodic boundary conditions, the eigenvectors of **K** have the form

$$e_a^\mu = \cos\left(\frac{2\pi\mu a}{N_v} - \phi\right) \tag{8.33}$$

for $\mu = 0, 1, 2, \ldots, N_v/2$, and with a phase parameter ϕ that can take any value from 0 to 2π. The eigenvalues are given by the discrete Fourier transform $\tilde{K}(\mu)$ of $K(|a - a'|)$, which is real in the case we consider (see the Mathematical Appendix). The principal eigenvector is the eigenfunction from equation 8.33 with μ value corresponding to the maximum of $\tilde{K}(\mu)$. The functions K and $\tilde{K}$ in figure 8.8 are each the difference of two Gaussian functions. $\tilde{K}$ has been plotted as a function of the spatial frequency $k = 2\pi\mu/(N_v d)$, where d is the cortical distance between locations a and $a + 1$. The value of μ corresponding to the principal eigenvector is determined by the k value corresponding to the maximum of the curve in figure 8.8B.

The oscillations in sign of the principal eigenvector, which is indicated by the dotted line in figure 8.8A, generate an alternating pattern of left- and right-eye innervation resembling the ocular dominance stripes seen in primary visual cortex (upper panel figure 8.7B). The lower panel of figure 8.7B shows the result of a simulation of Hebbian development of an ocular dominance map for a one-dimensional line across cortex consisting of 512 units. In this simulation, constraints that limit the growth of synaptic weights have been included, but these do not dramatically alter the conclusions of our analysis.

Competitive Hebb Rule

In this section and the next, we consider models that deal with Hebbian learning and development on a more abstract level. Although inspired by features of neural circuitry, these models are less directly linked to neurons and synapses. For example, the model discussed in this section considers

the actual spatial kernel $D_s(\vec{x})$ (figure 4.3B) are plotted under conditions of low and high noise. The linear kernels in figure 4.3B have been constructed by assuming that $\tilde{D}_s(\vec{\kappa})$ satisfies equation 4.47 and is real, which minimizes the spatial extent of the resulting receptive field. The resulting function $D_s(\vec{x})$ is radially symmetric, so it depends only on the distance $|\vec{x}|$ from the center of the receptive field to the point $\vec{x}$, and this radial dependence is plotted in figure 4.3B. Under low noise conditions (solid lines in figure 4.3), the linear kernel has a bandpass character and the predicted receptive field has a center-surround structure, which matches the retinal ganglion receptive fields shown in chapter 2. This structure eliminates one major source of redundancy in natural scenes: the strong similarity of neighboring inputs owing to the predominance of low spatial frequencies in images.

When the noise level is high (dashed lines in figure 4.3), the structure of the optimal receptive field is different. In spatial frequency terms, the filter is now low-pass, and the receptive field loses its surround. This structure averages over neighboring pixels to extract the true signal obscured by the uncorrelated noise. In the retina, we expect the signal-to-noise ratio to be controlled by the level of ambient light, with low levels of illumination corresponding to the high-noise case. The predicted change in the receptive fields at low illumination (high noise) matches what actually happens in the retina. At low light levels, circuitry changes within the retina remove the opposing surrounds from retinal ganglion cell receptive fields.

Temporal Processing in the LGN

Natural images tend to change relatively slowly over time. This means that there is substantial redundancy in the succession of natural images, suggesting an opportunity for efficient temporal filtering to complement efficient spatial filtering. An analysis similar to that of the previous section can be performed to account for the temporal receptive fields of visually responsive neurons early in the visual pathway. Recall that the predicted linear temporal response is given by $L_t(t)$, as expressed in equation 4.34. The analog of equation 4.37 for temporal decorrelation and variance equalization is

$$\langle L_t(t) L_t(t') \rangle = \sigma_L^2 \delta(t - t') \,. \tag{4.48}$$

This is mathematically identical to equation 4.37 except that the role of the spatial variables $\vec{a}$ and $\vec{b}$ has been replaced by the temporal variables t and t'. The analysis proceeds exactly as above, and the optimal filter is the product of a noise filter and a temporal whitening filter, as before. The temporal linear kernel $D_t(\tau)$ is written in terms of its Fourier transform,

$$D_t(\tau) = \frac{1}{2\pi} \int d\omega \, \exp(-i\omega\tau) \tilde{D}_t(\omega) \,, \tag{4.49}$$

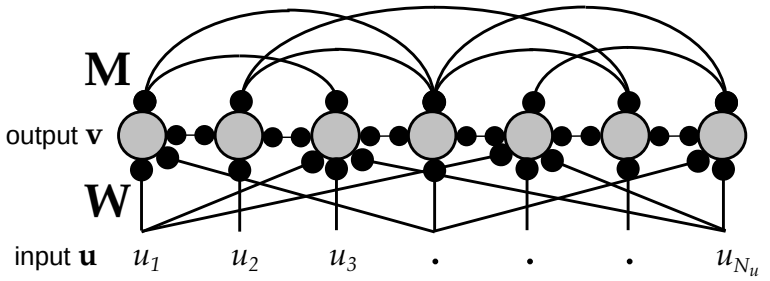

Figure 8.6 A network with multiple output units driven by feedforward synapses with weights **W**, and interconnected by recurrent synapses with weights **M**.

determined by

$$\mathbf{v} = \mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v} \,. \tag{8.29}$$

$\mathbf{K} = (\mathbf{I} - \mathbf{M})^{-1}$

Equation 8.29 can be solved by defining the matrix inverse $\mathbf{K} = (\mathbf{I} - \mathbf{M})^{-1}$, where **I** is the identity matrix, to obtain

$$\mathbf{v} = \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{u} \quad \text{or} \quad v_a = \sum_{a'=1}^{N_v} \sum_{b=1}^{N_u} K_{aa'} W_{a'b} u_b \,. \tag{8.30}$$

It is important for different output neurons in a multi-unit network to be selective for different aspects of the input, or else their responses will be redundant. For the case of a single cell, competition between different synapses could be used to ensure that synapse-specific plasticity rules do not modify all of the synapses onto a postsynaptic neuron in the same way. For multiple output networks, fixed or plastic linear or nonlinear recurrent interactions can be used to ensure that the units do not all develop the same selectivity. In the following sections, we consider three different patterns of plasticity: plastic feedforward and fixed recurrent synapses, plastic feedforward and recurrent synapses, and, finally, fixed feedforward and plastic recurrent synapses.

Hebbian Development of Ocular Dominance Stripes

Ocular dominance stripes can arise in a Hebbian model with plastic feedforward but fixed recurrent synapses. In the single-cell model of ocular dominance considered previously, the ultimate ocular preference of the output cell depends on the initial conditions of its synaptic weights. A multiple-output version of the model without any recurrent connections would therefore generate a random pattern of selectivities across the cortex if it started with random weights. Figure 8.7B shows that ocular dominance is actually organized in a highly structured cortical map. Dominance by the left and right eye alternates across the cortex, forming a striped pattern. Such a map can arise in the context of Hebbian development of feedforward weights if we include a fixed intracortical connection matrix **M**.

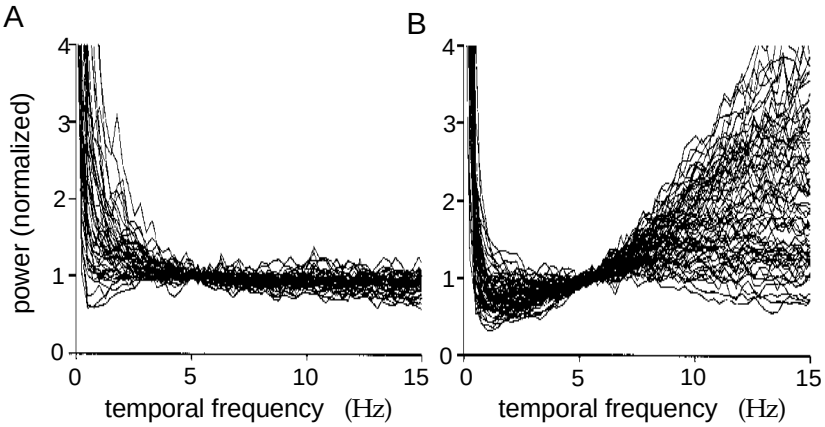

Figure 4.5 (A) Power spectra of the spike trains of 51 cat LGN cells in response to presentation of the movie *Casablanca*, normalized to their own values between 5 and 6 Hz. B) Equivalently normalized power spectra of the spike trains of 75 LGN cells in response to white-noise stimuli. (Adapted from Dan et al., 1996.)

spectra of spike trains of cat LGN cells in response to natural scenes (the movie *Casablanca*), and figure 4.5B shows power spectra in response to white-noise stimuli. The power spectra of the responses to natural scenes are quite flat above about $\omega = 3$ Hz. In response to white noise, on the other hand, they rise with ω. This is exactly what we would expect if LGN cells are acting as temporal whitening filters. In the case of natural stimuli, the whitening filter evenly distributes the output power over a broad frequency range. Responses to white-noise stimuli increase at high frequencies due to the boosting of inputs at these frequencies by the whitening filter.

Cortical Coding

Computational concerns beyond mere linear information transfer are likely to be relevant at the level of cortical processing of visual images. For one thing, the primary visual cortex has many more neurons than the LGN, yet they can collectively convey no more information about the visual world than they receive. As we saw in chapter 2, neurons in primary visual cortex are selective for quantities, such as spatial frequency and orientation, that are of particular importance in relation to object recognition but not for information transfer. Nevertheless, the methods described in the previous section can be used to understand restricted aspects of receptive fields of neurons in primary visual cortex, namely, the way that their multiple selectivities are collectively assigned. For example, cells that respond best at high spatial frequencies tend to respond more to low temporal frequency components of images, and vice versa.

The stimulus power spectrum written as a function of both spatial and temporal frequency has been estimated as $\tilde{Q}_{ss}(\vec{\kappa}, \omega) \propto 1/\left(|\vec{\kappa}|^2 + \alpha^2\omega^2\right)$,

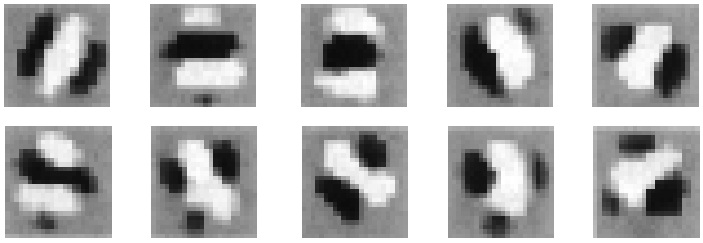

Figure 8.5 Different cortical receptive fields arising from a correlation-based developmental model. White and black regions correspond to areas in the visual field where ON-center (white regions) or OFF-center (black regions) LGN cells excite the cortical neuron being modeled. (Adapted from Miller, 1994.)

plasticity, subject to appropriate constraints, to the feedforward weights of the model.

As in the case of ocular dominance, the development of orientation selectivity can be partially analyzed on the basis of properties of the correlation matrix driving Hebbian development. The critical feature required to produce orientation-selective receptive fields is the growth of components proportional to eigenvectors that are not spatially uniform or rotationally invariant. Application of a Hebbian rule without constraints leads to growth of a uniform component resulting in an unstructured receptive field. Appropriate constraints can eliminate growth of this component, producing spatial structured receptive fields. The development of receptive fields that are not rotationally invariant, and that therefore exhibit orientation selectivity, relies on nonlinear aspects of the model and is therefore difficult to study analytically. For this reason, we simply present simulation results.

arbor function

Neurons in primary visual cortex receive afferents from LGN cells centered over a finite region of the visual field. This anatomical constraint can be included in developmental models by introducing what is called an arbor function. The arbor function, which is often taken to be Gaussian, characterizes the density of innervation from different visual locations to the cell being modeled. As a simplification, this density is not altered during the Hebbian developmental process, but the strengths of synapses within the arbor are modified by the Hebbian rule. The outcome is oriented receptive fields of a limited spatial extent. Figure 8.5 shows the weights resulting from a simulation of receptive-field development using a large array of ON- and OFF-center LGN afferents. This illustrates the variety of oriented receptive field structures that can arise through a Hebbian developmental rule.

Temporal Hebbian Rules and Trace Learning

Unlike correlation- or covariance-based rules, temporal Hebbian rules allow changes of synaptic strength to depend on the temporal sequence of

4.3 Entropy and Information for Spike Trains

Computing the entropy or information content of a neuronal response characterized by spike times is much more difficult than computing these quantities for responses described by firing rates. Nevertheless, these computations are important, because firing rates are incomplete descriptions that can lead to serious underestimates of the entropy and information. In this section, we discuss how the entropy and mutual information can be computed for spike trains. Extensive further discussion can be found in the book by Rieke et al. (1997).

Spike-train entropy calculations are typically based on the study of long-duration recordings consisting of many action potentials. The entropy or mutual information typically grows linearly with the length of the spike train being considered. For this reason, the entropy and mutual information of spike trains are reported as entropy or information rates. These are the total entropy or information divided by the duration of the spike train. We write the entropy rate as $\dot{H}$ rather than H. Alternatively, entropy and mutual information can be divided by the total number of action potentials and reported as bits per spike rather than bits per second.

entropy and information rates

To compute entropy and information rates for a spike train, we need to determine the probabilities that various temporal patterns of action potentials appear. These probabilities replace the factors $P[r]$ or $p[r]$ that occur when discrete or continuous firing rates are used to characterize a neural response. The temporal pattern of a group of action potentials can be specified by listing either the individual spike times or the sequence of intervals between successive spikes. The entropy and mutual information calculations we present are based on a spike-time description, but as an initial example we consider an approximate computation of entropy using interspike intervals.

The probability of an interspike interval falling in the range between τ and $\tau + \Delta\tau$ is given in terms of the interspike interval probability density by $p[\tau]\Delta\tau$. Because the interspike interval is a continuous variable, we must specify a resolution $\Delta\tau$ with which it is measured to define the entropy. If the different interspike intervals are statistically independent, the entropy associated with the interspike intervals in a spike train of average rate $\langle r \rangle$ and of duration T is the number of intervals, $\langle r \rangle T$, times the integral over τ of $-p[\tau]\log_2(p[\tau]\Delta\tau)$. The entropy rate is obtained by dividing this result by T, and the entropy per spike requires dividing by the number of spikes, $\langle r \rangle T$. The assumption of independent interspike intervals is critical for obtaining the spike-train entropy solely in terms of $p[\tau]$. Correlations between different interspike intervals reduce the total entropy, so the result obtained by assuming independent intervals provides an upper bound on the true entropy of a spike train. Thus, in general, the entropy rate $\dot{H}$ for a spike train with interspike interval distribution $p[\tau]$ and average rate $\langle r \rangle$

by noise that is also Gaussian, maximizing the variance of v by a Hebbian rule maximizes not only the output entropy but also the amount of information v carries about $\mathbf{u}$. In chapter 10, we further consider the computational significance of finding the direction of maximum variance in the input data set, and we discuss the relationship between this and general techniques for extracting structure from input statistics.

Figure 8.4B shows the consequence of applying correlation-based Hebbian plasticity when the average activities of the inputs are not 0, as is inevitable if real firing rates are employed. In this example, correlation-based Hebbian modification aligns the weight vector parallel to a line from the origin to the point $\langle \mathbf{u} \rangle$. This clearly fails to capture the essence of the distribution of inputs. Figure 8.4C shows the result of applying covariance-based Hebbian modification instead. Now the weight vector is aligned with the cloud of data points because this rule finds the direction of the principal eigenvector of the covariance matrix $\mathbf{C}$ of equation 8.11 rather the correlation matrix $\mathbf{Q}$.

Hebbian Development of Ocular Dominance

As an example of a developmental model of ocular dominance at the single neuron level, we consider the highly simplified case of a single layer 4 cell that receives input from just two LGN afferents. One afferent is associated with the right eye and has activity u_R, and the other is from the left eye and has activity u_L. Two synaptic weights $\mathbf{w} = (w_R, w_L)$ describe the strengths of these projections, and the output activity v is determined by equation 8.2,

$$v = w_R u_R + w_L u_L \,. \tag{8.24}$$

The weights in this model are constrained to nonnegative values. Initially, the weights for the right- and left-eye inputs are taken to be approximately equal. Ocular dominance arises when one of the weights is pushed to 0 while the other remains positive.

We can estimate the results of a Hebbian developmental process by considering the input correlation matrix. We assume that the two eyes are statistically equivalent, so the correlation matrix of the right- and left-eye inputs takes the form

$$\mathbf{Q} = \langle \mathbf{u}\mathbf{u} \rangle = \begin{pmatrix} \langle u_R u_R \rangle & \langle u_R u_L \rangle \\ \langle u_L u_R \rangle & \langle u_L u_L \rangle \end{pmatrix} = \begin{pmatrix} q_S & q_D \\ q_D & q_S \end{pmatrix} . \tag{8.25}$$

The subscripts S and D denote same- and different-eye correlations. The eigenvectors are $\mathbf{e}_1 = (1, 1)/\sqrt{2}$ and $\mathbf{e}_2 = (1, -1)/\sqrt{2}$ for this correlation matrix, and their eigenvalues are $\lambda_1 = q_S + q_D$ and $\lambda_2 = q_S - q_D$.

If the right- and left-eye weights evolve according to equation 8.5, it is straightforward to show that the eigenvector combinations $w_+ = w_R + w_L$

However, any correlations between successive intervals (if $B(t + T_s)$ is correlated with $B(t)$, for example) reduce the total spike-train entropy, causing equation 4.54 to overestimate the true entropy rate. Thus, for finite T_s, this equation provides an upper bound on the true entropy rate. If T_s is too small, $B(t + T_s)$ and $B(t)$ are likely to be correlated, and the overestimate may be severe. As T_s increases, we expect the correlations to get smaller, and equation 4.54 should provide a more accurate value. For any finite data set, T_s cannot be increased past a certain point, because there will not be enough spike sequences of duration T_s in the data set to determine their probabilities. Thus, in practice, T_s must be increased until the point where the extraction of probabilities becomes problematic, and some form of extrapolation to $T_s \to \infty$ must be made.

Statistical mechanics arguments suggest that the difference between the entropy for finite T_s and the true entropy for $T_s \to \infty$ should be proportional to $1/T_s$ for large T_s. Therefore, the true entropy can be estimated, as in figure 4.7, by linearly extrapolating a plot of the entropy rate versus $1/T_s$ to the point $1/T_s = 0$. In figure 4.7 (upper line), this has been done for data from the motion-sensitive H1 neuron of the fly visual system. The plotted points show entropy rates computed for different values of $1/T_s$, and they vary linearly over most of the range of the plot. However, when $1/T_s$ goes below about 20/s (or $T_s > 50$ ms), the dependence suddenly changes. This is the point at which the amount of data is insufficient to extract even an overestimate of the entropy. By linearly extrapolating the linear part of the series of computed points in figure 4.7, Strong et al. estimated that the H1 spike trains had an entropy rate of 157 bits/s when the resolution was $\Delta t = 3$ ms.

To compute the mutual information rate for a spike train, we must subtract the noise entropy rate from the full spike-train entropy rate. The noise entropy rate is determined from the probabilities of finding various sequences B, given that they were evoked by the same stimulus. This is done by considering sequences $B(t)$ that start at a fixed time t. If the same stimulus is used in repeated trials, sequences that begin at time t in every trial are generated by the same stimulus. Therefore, the conditional probability of the response, given the stimulus, is in this case the distribution $P[B(t)]$ for response sequences beginning at time t. This is obtained by determining the fraction of trials on which $B(t)$ was evoked. Note that $P[B(t)]$ is the probability of finding a given sequence at time t within a set of spike trains obtained on trials using the same stimulus. In contrast, $P[B]$, used in the spike-train entropy calculation, is the probability of finding the sequence B at any time within these trains. Determining $P[B(t)]$ for a sufficient number of spike sequences may take a large number of trials using the same stimulus.

The full noise entropy is computed by averaging the noise entropy at time t over all t values. The average over t plays the role of the average over

use the Oja rule, 8.16, instead of the basic Hebb rule. The weight vector generated by the Oja rule, in the example we have discussed, approaches $\mathbf{w} = \mathbf{e}_1/\alpha^{1/2}$ as $t \to \infty$. In other words, the Oja rule gives a weight vector that is parallel to the principal eigenvector, but normalized to a length of $1/\alpha^{1/2}$ rather than growing without bound.

averaged Hebb rule with subtractive constraint

Finally, we examine the effect of including a subtractive constraint, as in equation 8.14. Averaging equation 8.14 over the training inputs, we find

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q} \cdot \mathbf{w} - \frac{(\mathbf{w} \cdot \mathbf{Q} \cdot \mathbf{n})\mathbf{n}}{N_u} . \tag{8.22}$$

If we once again express $\mathbf{w}$ as a sum of eigenvectors of $\mathbf{Q}$, we find that the growth of each coefficient in this sum is unaffected by the extra term in equation 8.22, provided that $\mathbf{e}_\mu \cdot \mathbf{n} = 0$. However, if $\mathbf{e}_\mu \cdot \mathbf{n} \neq 0$, the extra term modifies the growth. In our discussion of ocular dominance, we consider a case in which the principal eigenvector of the correlation matrix is proportional to $\mathbf{n}$. In this case, $\mathbf{Q} \cdot \mathbf{e}_1 - (\mathbf{e}_1 \cdot \mathbf{Q} \cdot \mathbf{n})\mathbf{n}/N = 0$, so the projection in the direction of the principal eigenvector is unaffected by the synaptic plasticity rule. Furthermore, $\mathbf{e}_\mu \cdot \mathbf{n} = 0$ for $\mu \geq 2$ because the eigenvectors of the correlation matrix are mutually orthogonal, which implies that the evolution of the remaining eigenvectors is unaffected by the constraint. As a result,

$$\mathbf{w}(t) = (\mathbf{w}(0) \cdot \mathbf{e}_1)\,\mathbf{e}_1 + \sum_{\mu=2}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)(\mathbf{w}(0) \cdot \mathbf{e}_\mu)\,\mathbf{e}_\mu . \tag{8.23}$$

Thus, ignoring the effects of any possible saturation constraints, the synaptic weight matrix tends to become parallel to the eigenvector with the second largest eigenvalue as $t \to \infty$.

In summary, if weight growth is limited by some form of multiplicative normalization, as in the Oja rule, the configuration of synaptic weights produced by Hebbian modification typically will be proportional to the principal eigenvector of the input correlation matrix. When subtractive normalization is used and the principal eigenvector is proportional to $\mathbf{n}$, the eigenvector with the next largest eigenvalue provides an estimate of the configuration of final weights, again up to a proportionality factor. If, however, saturation constraints are used, as they must be in a subtractive scheme, this can invalidate the results of a simplified analysis based solely on these eigenvectors (as in figure 8.3). Nevertheless, we base our discussion of the Hebbian development of ocular dominance and cortical maps on an analysis of the eigenvectors of the input correlation matrix. We present simulation results to verify that this analysis is not invalidated by the constraints imposed in the full models.

Principal Component Projection

If applied for a long enough time, both the basic Hebb rule (equation 8.3) and the Oja rule (equation 8.16) generate weight vectors that are parallel

in the computation of the information rate, but the information can still depend on Δt through nondivergent terms. This reflects the fact that more information can be extracted from accurately measured spike times than from poorly measured spike times. Thus, we expect the information rate to increase with decreasing Δt, at least over some range of Δt values. At some critical value of Δt that matches the natural degree of noise jitter in the spike timings, we expect the information rate to stop increasing. This value of Δt is interesting because it tells us about the degree of spike timing accuracy in neural encoding.

The information conveyed by spike trains can be used to compare responses to different stimuli and thereby reveal stimulus-specific aspects of neural encoding. For example, Rieke et al. (1995) compared the information conveyed by single neurons in a peripheral auditory organ (the amphibian papilla) of the bullfrog in response to broadband noise or to noise filtered to have an amplitude spectrum close to that of natural bullfrog calls (although the phases for each frequency component were chosen randomly). They determined that the cells conveyed on average of 46 bits per second (1.4 bits per spike) for broadband noise and 133 bits per second (7.8 bits per spike) for stimuli with call-like spectra, despite the fact that the broadband noise had a higher entropy. The spike trains in response to the call-like stimuli conveyed information with near maximal efficiency.

4.4 Chapter Summary

Shannon's information theory can be used to determine how much a neural response tells both us and, presumably, the animal in which the neuron lives, about a stimulus. Entropy is a measure of the uncertainty or surprise associated with a stochastic variable, such as a stimulus. Mutual information quantifies the reduction in uncertainty associated with the observation of another variable, such as a response. The mutual information is related to the Kullback-Leibler divergence between two probability distributions. We defined the response and noise entropies for probability distributions of discrete and continuous firing rates, and considered how the information transmitted about a set of stimuli might be optimized. The principles of entropy and information maximization were used to account for features of the receptive fields of cells in the retina, LGN, and primary visual cortex. This analysis introduced probability factorization and equalization, and whitening and noise filters. Finally, we discussed how the information conveyed about dynamic stimuli by spike sequences can be estimated.

ocular dominance

Ocular dominance refers to the tendency of input to neurons in the adult primary visual cortex of many mammalian species to favor one eye over the other. This is especially true for neurons in layer 4, which receive extensive innervation from the LGN. Neurons dominated by one eye or the other occupy different patches of cortex, and areas with left- or right-eye ocular dominance alternate across the cortex in fairly regular bands known as ocular dominance stripes. In a later section, we discuss how this cortical map can arise from Hebbian plasticity.

ocular dominance stripes

Single Postsynaptic Neuron

Equation 8.5, which shows the consequence of averaging the basic Hebb rule over all the presynaptic training patterns, is a linear equation for $\mathbf{w}$. Provided that we ignore any constraints on $\mathbf{w}$, it can be analyzed using standard techniques for solving differential equations (see chapter 7 and the Mathematical Appendix). In particular, we use the method of matrix diagonalization, which involves expressing $\mathbf{w}$ in terms of the eigenvectors of $\mathbf{Q}$. These are denoted by $\mathbf{e}_\mu$ with $\mu = 1, 2, \cdots, N_u$, and they satisfy $\mathbf{Q} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu$. For correlation or covariance matrices, all the eigenvalues, λ_μ for all μ, are real and nonnegative, and for convenience we order them so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{N_u}$.

Any N_u-dimensional vector can be represented using the eigenvectors as a basis, so we can write

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu \,, \tag{8.19}$$

where the coefficients are equal to the dot products of $\mathbf{w}$ with the corresponding eigenvectors. For example, at time 0, $c_\mu(0) = \mathbf{w}(0) \cdot \mathbf{e}_\mu$. Writing $\mathbf{w}$ as a sum of eigenvectors turns matrix multiplication into ordinary multiplication, making calculations easier. Substituting the expansion 8.19 into 8.5 and following the procedure presented in chapter 7 for isolating uncoupled equations for the coefficients, we find that $c_\mu(t) = c_\mu(0)\exp(\lambda_\mu t/\tau_w)$. Going back to equation 8.19, this means that

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)\left(\mathbf{w}(0) \cdot \mathbf{e}_\mu\right)\mathbf{e}_\mu \,. \tag{8.20}$$

The exponential factors in 8.20 all grow over time, because the eigenvalues λ_μ are positive for all μ values. For large t, the term with the largest value of λ_μ (assuming it is unique) becomes much larger than any of the other terms and dominates the sum for $\mathbf{w}$. This largest eigenvalue has the label $\mu = 1$, and its corresponding eigenvector $\mathbf{e}_1$ is called the principal eigenvector. Thus, for large t, $\mathbf{w} \propto \mathbf{e}_1$ to a good approximation, provided that $\mathbf{w}(0) \cdot \mathbf{e}_1 \neq 0$. After training, the response to an arbitrary input vector $\mathbf{u}$ is well approximated by

principal eigenvector

$$v \propto \mathbf{e}_1 \cdot \mathbf{u} \,. \tag{8.21}$$

II Neurons and Neural Circuits

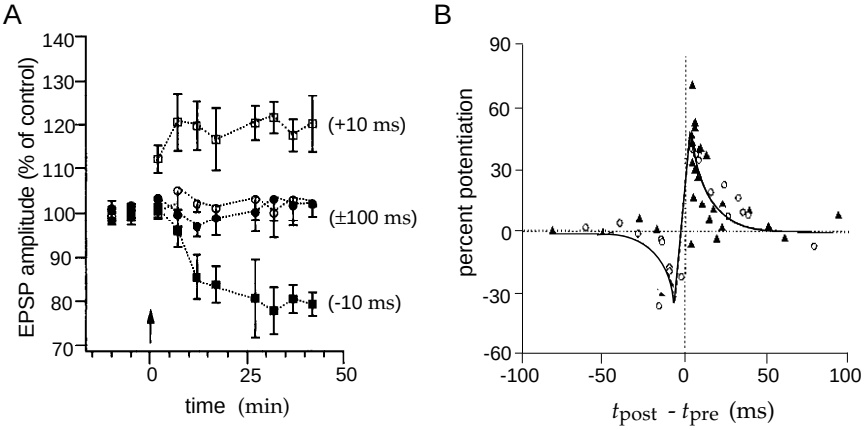

Figure 8.2 LTP and LTD produced by 50 to 75 pairs of pre- and postsynaptic action potential with various timings. (A) The amplitude of the excitatory postsynaptic potential (EPSP) evoked by stimulation of the presynaptic neuron plotted at various times as a percentage of the amplitude prior to paired stimulation. At the time indicated by the arrow, paired stimulations of the presynaptic and postsynaptic neurons were performed. For the traces marked by open symbols, the presynaptic spike occurred either 10 or 100 ms before the postsynaptic neuron fired an action potential. The traces marked by solid symbols denote the reverse ordering in which the presynaptic spike occurred either 10 or 100 ms after the postsynaptic spike. Separations of 100 ms had no long-lasting effect. In contrast, the 10 ms delays produced effects that built up to a maximum over a 5-to-10-minute period and lasted for the duration of the experiment. Pairing a presynaptic action potential with a postsynaptic action potential 10 ms later produced LTP, whereas the reverse ordering generated LTD. (B) LTP and LTD of retinotectal synapses recorded in vivo in *Xenopus* tadpoles. The percent change in synaptic strength produced by multiple pairs of action potentials is plotted as a function of their time difference. The filled symbols correspond to extracellular stimulation of the postsynaptic neuron, and the open symbols, to intracellular stimulation. The H function in equation 8.18 is proportional to the solid curve. (A adapted from Markram et al., 1997; B adapted from Zhang et al., 1998.)

τ between the times when the firing rates of the pre- and postsynaptic neurons are evaluated. A function $H(\tau)$ determines the rate of synaptic modification that occurs due to postsynaptic activity separated in time from presynaptic activity by an interval τ. The total rate of synaptic modification is determined by integrating over all time differences τ. If we assume that the rate of synaptic modification is proportional to the product of the pre- and postsynaptic rates, as it is for a Hebbian rule, the rate of change of the synaptic weights at time t is given by

timing-based rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty d\tau \; (H(\tau)v(t)\mathbf{u}(t-\tau) + H(-\tau)v(t-\tau)\mathbf{u}(t)) \; . \qquad (8.18)$$

If $H(\tau)$ is positive for positive τ and negative for negative τ, the first term on the right side of this equation represents LTP, and the second, LTD. The solid curve in figure 8.2B is an example of such an H function. The temporal asymmetry of H has a dramatic effect on synaptic weight changes because it causes them to depend on the temporal order of the pre- and

5 Model Neurons I: Neuroelectronics

5.1 Introduction

A great deal is known about the biophysical mechanisms responsible for generating neuronal activity, and this knowledge provides a basis for constructing neuron models. Such models range from highly detailed descriptions involving thousands of coupled differential equations to greatly simplified caricatures useful for studying large interconnected networks. In this chapter, we discuss the basic electrical properties of neurons and the mathematical models by which they are described. We present a simple but nevertheless useful model neuron, the integrate-and-fire model, in a basic version and with added membrane and synaptic conductances. We also discuss the Hodgkin-Huxley model, which describes the conductances responsible for generating action potentials. In chapter 6, we continue by presenting more complex models, in terms of their conductances and morphology. Circuits and networks of model neurons are discussed in chapter 7. This chapter makes use of basic concepts of electrical circuit theory, which are reviewed in the Mathematical Appendix.

5.2 Electrical Properties of Neurons

Like other cells, neurons are packed with a huge number and variety of ions and molecules. A cubic micron of cytoplasm might contain, for example, 10^{10} water molecules, 10^8 ions, 10^7 small molecules such as amino acids and nucleotides, and 10^5 proteins. Many of these molecules carry charges, either positive or negative. Most of the time, there is an excess concentration of negative charge inside a neuron. Excess charges that are mobile, like ions, repel each other and build up on the inside surface of the cell membrane. Electrostatic forces attract an equal density of positive ions from the extracellular medium to the outside surface of the membrane.

cell membrane

The cell membrane is a lipid bilayer 3 to 4 nm thick that is essentially impermeable to most charged molecules. This insulating feature causes the cell membrane to act as a capacitor by separating the charges lying

Subtractive Normalization

The sum over synaptic weights that is constrained by subtractive normalization can be written as $\sum w_b = \mathbf{n} \cdot \mathbf{w}$ where $\mathbf{n}$ is an N_u-dimensional vector with all its components equal to 1 (as introduced in chapter 7). This sum can be constrained by replacing equation 8.3 with

Hebb rule with subtractive normalization

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \frac{v(\mathbf{n} \cdot \mathbf{u})\mathbf{n}}{N_u} . \tag{8.14}$$

Note that $\mathbf{n} \cdot \mathbf{u}$ is simply the sum of all the inputs. This rule imposes what is called subtractive normalization because the same quantity is subtracted from the change to each weight, whether that weight is large or small. Subtractive normalization imposes the constraint on the sum of weights rigidly because it does not allow the Hebbian term to change $\mathbf{n} \cdot \mathbf{w}$. To see this, we take the dot product of equation 8.14 with $\mathbf{n}$ to obtain

$$\tau_w \frac{d\mathbf{n} \cdot \mathbf{w}}{dt} = v\mathbf{n} \cdot \mathbf{u} \left(1 - \frac{\mathbf{n} \cdot \mathbf{n}}{N_u}\right) = 0 . \tag{8.15}$$

The last equality follows because $\mathbf{n} \cdot \mathbf{n} = N_u$. Hebbian modification with subtractive normalization is nonlocal in that it requires the sum of all the input activities, $\mathbf{n} \cdot \mathbf{u}$, to be available to the mechanism that modifies each synapse. It is not obvious how such a rule could be implemented biophysically.

Subtractive normalization must be augmented by a saturation constraint that prevents weights from becoming negative. If the rule 8.14 attempts to drive any of the weights below 0, the saturation constraint prevents this change. At this point, the rule is not applied to any saturated weights, and its effect on the other weights is modified. Both modifications can be achieved by setting the components of the vector $\mathbf{n}$ corresponding to any saturated weights to 0 and the factor of N_u in equation 8.14 equal to the sum of the components of this modified $\mathbf{n}$ vector (i.e., the number of unsaturated components). Without any upper saturation limit, this procedure often results in a final outcome in which all the weights but one have been set to 0. To avoid this, an upper saturation limit is also typically imposed. Hebbian plasticity with subtractive normalization is highly competitive because small weights are reduced by a larger proportion of their sizes than are large weights.

Multiplicative Normalization and the Oja Rule

A constraint on the sum of the squares of the synaptic weights can be imposed dynamically by using a modification of the basic Hebb rule known as the Oja rule (Oja, 1982),

Oja rule

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w} , \tag{8.16}$$

is the Boltzmann constant and T is the temperature on an absolute Kelvin scale. For chemists and biologists (though not for physicists), it is more customary to discuss moles of ions rather than single ions. A mole of ions has Avogadro's number times as much thermal energy as a single ion, or RT, where R is the universal gas constant, equal to 8.31 joules/mol K° = 1.99 cal/mol K°. RT is about 2500 joules/mol or 0.6 kCal/mol at normal temperatures.

To estimate the size of typical membrane potentials, we equate the thermal energy of a mole of ions to the energy gained or lost when a mole of ions crosses a membrane with a potential difference V_T across it. This energy is FV_T, where F is the Faraday constant, F = 96,480 coulombs/mol, equal to Avogadro's number times the charge of a single proton, q. Setting $FV_T = RT$ gives

V_T

$$V_T = \frac{RT}{F} = \frac{k_B T}{q}. \tag{5.1}$$

This is an important parameter that enters into a number of calculations. V_T is between 24 and 27 mV for the typical temperatures of cold- and warm-blooded animals. This sets the overall scale for membrane potentials across neuronal membranes, which range from about -3 to +2 times V_T.

Intracellular Resistance

Membrane potentials measured at different places within a neuron can take different values. For example, the potentials in the soma, dendrite, and axon can all be different. Potential differences between different parts of a neuron cause ions to flow within the cell, which tends to equalize these differences. The intracellular medium provides a resistance to such flow. This resistance is highest for long, narrow stretches of dendritic or axonal cable, such as the segment shown in figure 5.2. The longitudinal current I_L flowing along such a cable segment can be computed from Ohm's law. For the cylindrical segment of dendrite shown in figure 5.2, the longitudinal current flowing from right to left satisfies $V_2 - V_1 = I_L R_L$. Here, R_L is the longitudinal resistance, which grows in proportion to the length of the segment (long segments have higher resistances than short ones) and is inversely proportional to the cross-sectional area of the segment (thin segments have higher resistances than fat ones). The constant of proportionality, called the intracellular resistivity, r_L, typically falls in a range from 1 to 3 kΩ mm. The longitudinal resistance of the segment in figure 5.2 is r_L times the length L divided by the cross-sectional area πa^2, $R_L = r_L L/\pi a^2$. A segment 100 μm long with a radius of 2 μm has a longitudinal resistance of about 8 MΩ. A voltage difference of 8 mV would be required to force 1 nA of current down such a segment.

longitudinal current I_L

longitudinal resistance R_L

intracellular resistivity r_L

We can also use the intracellular resistivity to estimate crudely the conductance of a single channel. The conductance, being the inverse of a resistance, is equal to the cross-sectional area of the channel pore divided by

average over training inputs to obtain an averaged form of the plasticity rule. When the thresholds are set to their corresponding activity averages, equations 8.8 and 8.9 both produce the same averaged rule,

input covariance matrix **C**

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{C} \cdot \mathbf{w}\,, \tag{8.10}$$

where **C** is the input covariance matrix,

$$\mathbf{C} = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)(\mathbf{u} - \langle \mathbf{u} \rangle) \rangle = \langle \mathbf{u}\mathbf{u} \rangle - \langle \mathbf{u} \rangle^2 = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)\mathbf{u} \rangle\ . \tag{8.11}$$

covariance rules

Because of the presence of the covariance matrix in equation 8.10, equations 8.8 and 8.9 are known as covariance rules.

homosynaptic and heterosynaptic depression

Although they both average to give equation 8.10, the rules in equations 8.8 and 8.9 have their differences. Equation 8.8 modifies synapses only if they have nonzero presynaptic activities. When $v < \theta_v$, this produces an effect called homosynaptic depression. In contrast, equation 8.9 reduces the strengths of inactive synapses if $v > 0$. This is called heterosynaptic depression. Note that maintaining $\theta_v = \langle v \rangle$ in equation 8.8 requires changing θ_v as the weights are modified. In contrast, the threshold in equation 8.9 is independent of the weights and does not need to be changed during the training period to keep $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$.

Even though covariance rules include LTD and thus allow weights to decrease, they are unstable because of the same positive feedback that makes the basic Hebb rule unstable. For either rule 8.8 with $\theta_v = \langle v \rangle$ or rule 8.9 with $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$, $\tau_w d|\mathbf{w}|^2/dt = 2v(v - \langle v \rangle)$. The time average of the right side of this equation is proportional to the variance of the output, $\langle v^2 \rangle - \langle v \rangle^2$, which is positive except in the trivial case when v is constant. Also similar to the case of the Hebb rule is the fact that the covariance rules are noncompetitive, but competition can be introduced by allowing the thresholds to slide, as described below.

The BCM Rule

The covariance-based rule of equation 8.8 does not require any postsynaptic activity to produce LTD, and rule 8.9 can produce LTD without presynaptic activity. Bienenstock, Cooper, and Munro (1982) suggested an alternative plasticity rule, for which there is experimental evidence, that requires both pre- and postsynaptic activity to change a synaptic weight. This rule, which is called the BCM rule, takes the form

BCM rule

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}\,(v - \theta_v)\ . \tag{8.12}$$

As in equation 8.8, θ_v acts as a threshold on the postsynaptic activity that determines whether synapses are strengthened or weakened.

If the threshold θ_v is held fixed, the BCM rule, like the basic Hebbian rule, is unstable. Synaptic modification can be stabilized against unbounded

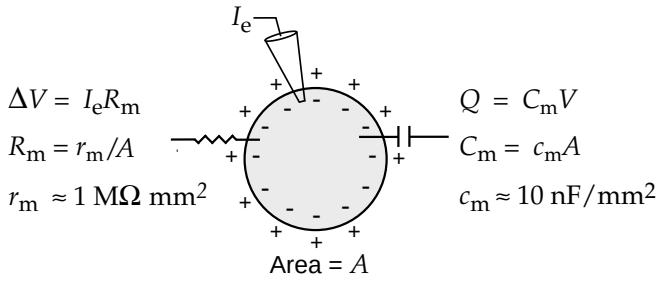

Figure 5.3 The capacitance and membrane resistance of a neuron considered as a single compartment. The membrane capacitance determines how the membrane potential V and excess internal charge Q are related. The membrane resistance R_m determines the size of the membrane potential deviation ΔV caused by a small current I_e entering through an electrode, for example. Equations relating the total membrane capacitance and resistance, C_m and R_m, to the specific membrane capacitance and resistance, c_m and r_m, are given along with typical values of c_m and r_m. The value of r_m may vary considerably under different conditions and for different neurons.

to 1 nF. For a neuron with a total membrane capacitance of 1 nF, 7×10^{-11} coulomb or about 10^9 singly charged ions are required to produce a resting potential of -70 mV. This is about 1/100,000 of the total number of ions in a neuron and is the amount of charge delivered by a 0.7 nA current in 100 ms.

We can use the membrane capacitance to determine how much current is required to change the membrane potential at a given rate. The time derivative of the basic equation relating the membrane potential and charge,

$$C_m \frac{dV}{dt} = \frac{dQ}{dt}, \tag{5.2}$$

plays an important role in the mathematical modeling of neurons. The time derivative of the charge dQ/dt is equal to the current passing into the cell, so the amount of current needed to change the membrane potential of a neuron with a total capacitance C_m at a rate dV/dt is $C_m dV/dt$. For example, 1 nA will change the membrane potential of a neuron with a capacitance of 1 nF at a rate of 1 mV/ms.

The capacitance of a neuron determines how much current is required to make the membrane potential change at a given rate. Holding the membrane potential steady at a level different from its resting value also requires current, but this current is determined by the membrane resistance rather than by the capacitance of the cell. For example, if a small constant current I_e is injected into a neuron through an electrode, as in figure 5.3, the membrane potential will shift away from its resting value by an amount ΔV given by Ohm's law, $\Delta V = I_e R_m$. R_m is known as the membrane or input resistance. The restriction to small currents and small ΔV is required because membrane resistances can vary as a function of voltage, whereas Ohm's law assumes R_m is constant over the range ΔV.

membrane resistance R_m

The Basic Hebb Rule

The simplest plasticity rule that follows the spirit of Hebb's conjecture takes the form

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}\,, \tag{8.3}$$

basic Hebb rule

where τ_w is a time constant that controls the rate at which the weights change. This equation, which we call the basic Hebb rule, implies that simultaneous pre- and postsynaptic activity increases synaptic strength. If the activity variables represent firing rates, the right side of this equation can be interpreted as a measure of the probability that the pre- and postsynaptic neurons both fire spikes during a small time interval.

Synaptic plasticity is generally modeled as a slow process that gradually modifies synaptic weights over a time period during which the components of $\mathbf{u}$ take a variety of different values. Each different set of $\mathbf{u}$ values is called an input pattern. The direct way to compute the weight changes induced by a series of input patterns is to sum the small changes caused by each of them separately. A convenient alternative is to average over all of the different input patterns and compute the weight changes induced by this average. As long as the synaptic weights change slowly enough, the averaging method provides a good approximation of the weight changes produced by the set of input patterns.

In this chapter, we use angle brackets $\langle\ \rangle$ to denote averages over the ensemble of input patterns presented during training (which is a slightly different usage from earlier chapters). The Hebb rule of equation 8.3, when averaged over the inputs used during training, becomes

averaged Hebb rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u}\rangle\,. \tag{8.4}$$

In unsupervised learning, v is determined by equation 8.2, and if we replace v with $\mathbf{w}\cdot\mathbf{u}$, we can rewrite the averaged plasticity rule (equation 8.4) as

correlation-based rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\cdot\mathbf{w} \quad \text{or} \quad \tau_w \frac{dw_b}{dt} = \sum_{b'=1}^{N_u} Q_{bb'} w_{b'}\,, \tag{8.5}$$

input correlation matrix $\mathbf{Q}$

where $\mathbf{Q}$ is the input correlation matrix given by

$$\mathbf{Q} = \langle \mathbf{u}\mathbf{u}\rangle \quad \text{or} \quad Q_{bb'} = \langle u_b u_{b'}\rangle\,. \tag{8.6}$$

Equation 8.5 is called a correlation-based plasticity rule because of the presence of the input correlation matrix.

Whether or not the pre- and postsynaptic activity variables are restricted to nonnegative values, the basic Hebb rule is unstable. To show this, we consider the square of the length of the weight vector, $|\mathbf{w}|^2 = \mathbf{w}\cdot\mathbf{w} = \sum_b w_b^2$. Taking the dot product of equation 8.3 with $\mathbf{w}$, and noting that $d|\mathbf{w}|^2/dt =$

tial. If the ion has an electric charge zq, where q is the charge of one proton, it must have a thermal energy of at least $-zqV$ to cross the membrane (this is a positive energy for $z > 0$ and $V < 0$). The probability that an ion has a thermal energy greater than or equal to $-zqV$, when the temperature (on an absolute scale) is T, is $\exp(zqV/k_B T)$. This is determined by integrating the Boltzmann distribution for energies greater than or equal to $-zqV$. In molar units, this result can be written as $\exp(zFV/RT)$, which is equal to $\exp(zV/V_T)$ by equation 5.1.

The biasing effect of the electrical potential can be overcome by an opposing concentration gradient. A concentration of ions inside the cell, [inside], that is sufficiently greater than the concentration outside the cell, [outside], can compensate for the Boltzmann probability factor. The rate at which ions flow into the cell is proportional to [outside]. The flow of ions out of the cell is proportional to [inside] times the Boltzmann factor, because in this direction only those ions that have sufficient thermal energy can leave the cell. The net flow of ions will be 0 when the inward and outward flows are equal. We use the letter E to denote the particular potential that satisfies this balancing condition, which is then

$$[\text{outside}] = [\text{inside}] \exp(zE/V_T) . \tag{5.3}$$

Solving this equation for E, we find *Nernst equation*

$$E = \frac{V_T}{z} \ln\left(\frac{[\text{outside}]}{[\text{inside}]}\right) . \tag{5.4}$$

Equation 5.4 is the Nernst equation. The reader can check that if the result is derived for either sign of ionic charge or membrane potential, the result is identical to 5.4, which thus applies in all cases. The equilibrium potential for a K^+ conducting channel, labeled E_K, typically falls in the range between -70 and -90 mV. The Na^+ equilibrium potential, E_{Na}, is 50 mV or higher, and E_{Ca}, for Ca^{2+} channels, is higher still, around 150 mV. Finally, Cl^- equilibrium potentials are typically around -60 to -65 mV, near the resting potential of many neurons.

The Nernst equation (5.4) applies when the channels that generate a particular conductance allow only one type of ion to pass through them. Some channels are not so selective, and in this case the potential E is not determined by equation 5.4. Instead, it takes a value intermediate between the equilibrium potentials of the individual ion types that it conducts. An approximate formula, known as the Goldman equation (see Tuckwell, 1988; or Johnston and Wu, 1995), can be used to estimate E for such conductances. In this case, E is often called a reversal potential, rather than an equilibrium potential, because the direction of current flow through the channel switches as the membrane potential passes through E. *Goldman equation* *reversal potential*

A conductance with an equilibrium or reversal potential E tends to move the membrane potential of the neuron toward the value E. When $V > E$, this means that positive current will flow outward, and when $V < E$, positive current will flow inward. Because Na^+ and Ca^{2+} conductances have

discuss their extension to supervised learning. As an example, we discuss the development of ocular dominance in cells of the primary visual cortex, and the formation of the map of ocular dominance preferences across the cortical surface. In the models we discuss, synaptic strengths are characterized by synaptic weights, defined as in chapter 7.

Stability and Competition

Increasing synaptic strength in response to activity is a positive feedback process. The activity that modifies synapses is reinforced by Hebbian plasticity, which leads to more activity and further modification. Without appropriate adjustments of the synaptic plasticity rules or the imposition of constraints, Hebbian modification tends to produce uncontrolled growth of synaptic strengths.

The easiest way to control synaptic strengthening is to impose an upper limit on the value that a synaptic weight can take. Such an upper limit is supported by LTP experiments. It also makes sense to prevent weights from changing sign, because the plasticity processes we are modeling cannot change an excitatory synapse into an inhibitory synapse or vice versa. We therefore impose the constraint, which we call a saturation constraint, that all excitatory synaptic weights must lie between 0 and a maximum value $w_{\max}$, which is a constant. The simplest implementation of saturation is to set any weight that would cross a saturation bound due to application of a plasticity rule to the limiting value.

synaptic saturation

Uncontrolled growth is not the only problem associated with Hebbian plasticity. Synapses are modified independently under a Hebbian rule, which can have deleterious consequences. For example, all of the synaptic weights may be driven to their maximum allowed values $w_{\max}$, causing the postsynaptic neuron to lose selectivity to different patterns of input. The development of input selectivity typically requires competition between different synapses, so that some are forced to weaken when others become strong. We discuss a variety of synaptic plasticity rules that introduce competition between synapses. In some cases, the same mechanism that leads to competition also stabilizes growth of the synaptic weights. In other cases, it does not, and saturation constraints must also be imposed.

synaptic competition

8.2 Synaptic Plasticity Rules

Rules for synaptic modification take the form of differential equations describing the rate of change of synaptic weights as a function of the pre- and postsynaptic activity and other possible factors. In this section, we give examples of such rules. In later sections, we discuss their computational implications.

called the leakage current. The currents carried by ion pumps that maintain the concentration gradients that make equilibrium potentials nonzero typically fall into this category. For example, one type of pump uses the energy of ATP hydrolysis to move three Na^+ ions out of the cell for every two K^+ ions it moves in. *leakage current*

It is normally assumed that ion pumps work at relatively steady rates so that the currents they generate can be included in a time-independent leakage conductance. Sometimes, this assumption is dropped and explicit pump currents are modeled. In either case, all of the time-independent contributions to the membrane current can be lumped together into a single leakage term $\overline{g}_L(V - E_L)$. Because this term hides many sins, its reversal potential E_L is not usually equal to the equilibrium potential of any specific ion. Instead, it is often kept as a free parameter and adjusted to make the resting potential of the model neuron match that of the cell being modeled. Similarly, $\overline{g}_L$ is adjusted to match the membrane conductance at rest. The line over the parameter $\overline{g}_L$ is used to indicate that it has constant value. A similar notation is used later in this chapter to distinguish variable conductances from the fixed parameters that describe them. The leakage conductance is called a passive conductance to distinguish it from variable conductances that are termed active. *resting potential*

5.3 Single-Compartment Models

Models that describe the membrane potential of a neuron by a single variable V are called single-compartment models. This chapter deals exclusively with such models. Multi-compartment models, which can describe spatial variations in the membrane potential, are considered in chapter 6. The equations for single-compartment models, like those of all neuron models, describe how charges flow into and out of a neuron and affect its membrane potential.

Equation 5.2 provides the basic relationship that determines the membrane potential for a single-compartment model. This equation states that the rate of change of the membrane potential is proportional to the rate at which charge builds up inside the cell. The rate of charge buildup is, in turn, equal to the total amount of current entering the neuron. The relevant currents are those arising from all the membrane and synaptic conductances plus, in an experimental setting, any current injected into the cell through an electrode. From equation 5.2, the sum of these currents is equal to $C_m dV/dt$, the total capacitance of the neuron times the rate of change of the membrane potential. Because the membrane current is usually characterized as a current per unit area, i_m, it is more convenient to divide this relationship by the surface area of the neuron. Then, the total current per unit area is equal to $c_m dV/dt$, where $c_m = C_m/A$ is the specific membrane capacitance. One complication in this procedure is that the electrode current, I_e, is not typically expressed as a current per unit area, so we must divide it by the total surface area of the neuron, A. Putting all

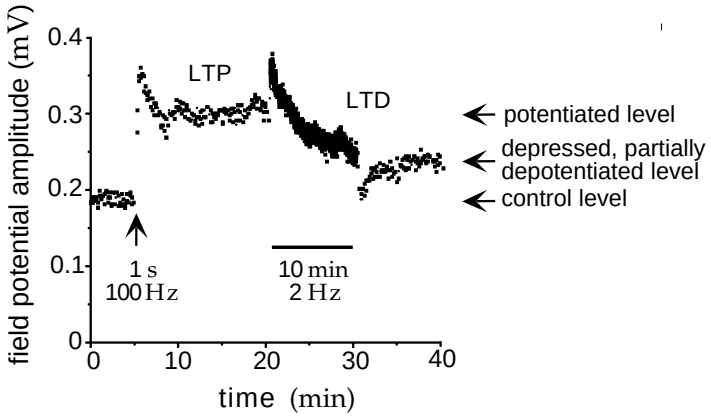

Figure 8.1 LTP and LTD at the Schaffer collateral inputs to the CA1 region of a rat hippocampal slice. The points show the amplitudes of field potentials evoked by constant amplitude stimulation. At the time marked by the arrow (at time 5 minutes), stimulation at 100 Hz for 1 s caused a significant increase in the response amplitude. Some of this increase decayed away following the stimulation, but most of it remained over the following 15 min test period, indicating LTP. Next, stimulation at 2 Hz was applied for 10 min (between times 20 and 30 minutes). This reduced the amplitude of the response. After a transient dip, the response amplitude remained at a reduced level approximately midway between the original and post-LTP values, indicating LTD. The arrows at the right show the levels initially (control), after LTP (potentiated), and after LTD (depressed, partially depotentiated). (Unpublished data of J. Fitzpatrick and J. Lisman.)

indicate amplitudes of field potentials evoked in the CA1 region of a slice of rat hippocampus by stimulation of the Schaffer collateral afferents. In experiments such as this, field potential amplitudes (or more often slopes) are used as a measure of synaptic strength. In figure 8.1, high-frequency stimulation induced synaptic potentiation (an increase in strength), and then long-lasting, low-frequency stimulation resulted in synaptic depression (a decrease in strength) that partially removed the effects of the previous potentiation. This is in accord with a generalized Hebb rule because high-frequency presynaptic stimulation evokes a postsynaptic response, whereas low-frequency stimulation does not. Changes in synaptic strength involve both transient and long-lasting effects, as seen in figure 8.1. Changes that persist for tens of minutes or longer are generally called long-term potentiation (LTP) and long-term depression (LTD). The longest-lasting forms appear to require protein synthesis.

potentiation

depression

LTP and LTD

A wealth of data is available on the underlying cellular basis of activity-dependent synaptic plasticity. For instance, the postsynaptic concentration of calcium ions appears to play a critical role in the induction of both LTP and LTD. However, we will not consider mechanistic models. Rather, we study synaptic plasticity at a functional level, attempting to relate the impact of synaptic plasticity on neurons and networks to the basic rules governing its induction.

the model neuron reaches a threshold value V_{th}. After the action potential, the potential is reset to a value V_{reset} below the threshold potential, $V_{\text{reset}} < V_{\text{th}}$.

The basic integrate-and-fire model was proposed by Lapicque in 1907, long before the mechanisms that generate action potentials were understood. Despite its age and simplicity, the integrate-and-fire model is still an extremely useful description of neuronal activity. By avoiding a biophysical description of the action potential, integrate-and-fire models are left with the simpler task of modeling only subthreshold membrane potential dynamics. This can be done with various levels of rigor. In the simplest version of these models, all active membrane conductances are ignored, including, for the moment, synaptic inputs, and the entire membrane conductance is modeled as a single passive leakage term, $i_{\text{m}} = \overline{g}_{\text{L}}(V - E_{\text{L}})$. This version is called the passive or leaky integrate-and-fire model. For small fluctuations about the resting membrane potential, neuronal conductances are approximately constant, and the passive integrate-and-fire model assumes that this constancy holds over the entire subthreshold range. For some neurons this is a reasonable approximation, and for others it is not. With these approximations, the model neuron behaves like an electric circuit consisting of a resistor and a capacitor in parallel (figure 5.4), and the membrane potential is determined by equation 5.6 with $i_{\text{m}} = \overline{g}_{\text{L}}(V - E_{\text{L}})$,

passive integrate-and-fire model

$$c_{\text{m}}\frac{dV}{dt} = -\overline{g}_{\text{L}}(V - E_{\text{L}}) + \frac{I_{\text{e}}}{A}\,. \tag{5.7}$$

It is convenient to multiply equation 5.7 by the specific membrane resistance r_{m}, which in this case is given by $r_{\text{m}} = 1/\overline{g}_{\text{L}}$. This cancels the factor of $\overline{g}_{\text{L}}$ on the right side of the equation and leaves a factor $c_{\text{m}}r_{\text{m}} = \tau_{\text{m}}$ on the left side, where τ_{m} is the membrane time constant of the neuron. The electrode current ends up being multiplied by r_{m}/A, which is the total membrane resistance R_{m}. Thus, the basic equation of the passive integrate-and-fire models is

$$\tau_{\text{m}}\frac{dV}{dt} = E_{\text{L}} - V + R_{\text{m}}I_{\text{e}}\,. \tag{5.8}$$

To generate action potentials in the model, equation 5.8 is augmented by the rule that whenever V reaches the threshold value V_{th}, an action potential is fired and the potential is reset to V_{reset}. Equation 5.8 indicates that when $I_{\text{e}} = 0$, the membrane potential relaxes exponentially with time constant τ_{m} to $V = E_{\text{L}}$. Thus, E_{L} is the resting potential of the model cell.

The membrane potential for the passive integrate-and-fire model is determined by integrating equation 5.8 (a numerical method for doing this is described in appendix A) and applying the threshold and reset rule for action potential generation. The response of a passive integrate-and-fire model neuron to a time-varying electrode current is shown in figure 5.5.

The firing rate of an integrate-and-fire model in response to a constant injected current can be computed analytically. When I_{e} is independent of

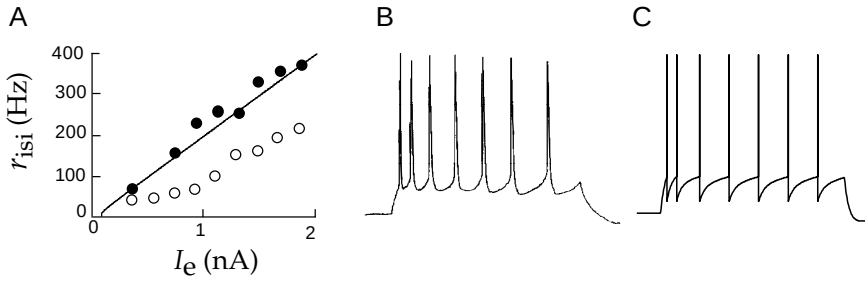

Figure 5.6 (A) Comparison of interspike-interval firing rates as a function of injected current for an integrate-and-fire model and a cortical neuron measure in vivo. The line gives r_{isi} for a model neuron with $\tau_m = 30$ ms, $E_L = V_{reset} = -65$ mV, $V_{th} = -50$ mV, and $R_m = 90$ MΩ. The data points are from a pyramidal cell in the primary visual cortex of a cat. The filled circles show the inverse of the interspike interval for the first two spikes fired, and the open circles show the steady-state interspike-interval firing rate after spike-rate adaptation. (B) A recording of the firing of a cortical neuron under constant current injection, showing spike-rate adaptation. (C) Membrane voltage trajectory and spikes for an integrate-and-fire model with an added current, with $r_m \Delta g_{sra}$ = 0.06, τ_{sra} = 100 ms, and E_K = -70 mV (see equations 5.13 and 5.14). (Data in A from Ahmed et al., 1998; B from McCormick, 1990.)

inverse of the interval between pairs of spikes. The rates determined in this way, using the first two spikes fired by the neuron in response to the injected current (filled circles in figure 5.6A), agree fairly well with the results of the integrate-and-fire model described in the figure caption. However, the real neuron exhibits spike-rate adaptation, in that the interspike intervals lengthen over time when a constant current is injected into the cell (figure 5.6B), before settling to a steady-state value. The steady-state firing rate in figure 5.6A (open circles) could also be fitted by an integrate-and-fire model, but not using the same parameters that were used to fit the initial spikes. Spike-rate adaptation is a common feature of cortical pyramidal cells, and consideration of this phenomenon allows us to show how an integrate-and-fire model can be modified to incorporate more complex dynamics.

spike-rate adaptation

Spike-Rate Adaptation and Refractoriness

The passive integrate-and-fire model that we have described thus far is based on two separate approximations, a highly simplified description of the action potential and a linear approximation for the total membrane current. If details of the action-potential generation process are not important for a particular modeling goal, the first approximation can be retained while the membrane current is modeled in as much detail as is necessary. We will illustrate this process by developing a heuristic description of spike-rate adaptation using a model conductance that has characteristics similar to measured neuronal conductances known to play important roles in producing this effect.

Seung's (1996) analysis of neural integration for eye position (see also Seung et al., 2001), which builds on Robinson (1989). In general, we followed Seung (1996) and Zhang (1996) in adopting the theoretical context of continuous line or surface attractors.

Sompolinsky & Shapley 1997 (see also Somers, Nelson & Sur, 1995; Carandini & Ringach, 1997) reviews the debate about the relative roles of feedforward and recurrent input as the source of orientation selectivity in primary visual cortex. We presented a model of a hypercolumn; an extension to multiple hypercolumns is used by Li (1998, 1999) to link psychophysical and physiological data on contour integration and texture segmentation. Persistent activity in prefrontal cortex during short-term memory tasks is reviewed by Goldman-Rakic (1994) and Fuster (1995) and is modeled by Compte et al. (2000).

Network associative memories were described and analyzed in Hopfield (1982; 1984) and Cohen & Grossberg (1983), where a general Lyapunov function is introduced. Grossberg (1988), **Amit (1989)**, and **Hertz, et al. (1991)** present theoretical results concerning associative networks, in particular their capacity to store information. Associative memory in nonbinary recurrent networks has been studied in particular by Treves and collaborators (see Rolls & Treves, 1998) and, in the context of line attractor networks, in Samsonovich & McNaughton (1997) and Battaglia & Treves (1998).

Rinzel and Ermentrout (1998) discusses phase-plane methods, and `XPP` (see `http://www.pitt.edu/~phase`) provides a computer environment for performing phase-plane and other forms of mathematical analysis on neuron and network models. We followed Li's (1995) presentation of Li & Hopfield's (1989) oscillatory model of the olfactory bulb.

The Boltzmann machine was introduced by Hinton & Sejnowski (1986) as a stochastic generalization of the Hopfield network (Hopfield, 1982). The mean-field model is due to Hopfield (1984), and we followed the probabilistic discussion given in Jordan et al. (1998). Markov chain methods for performing probabilistic inference are discussed in Neal (1993).

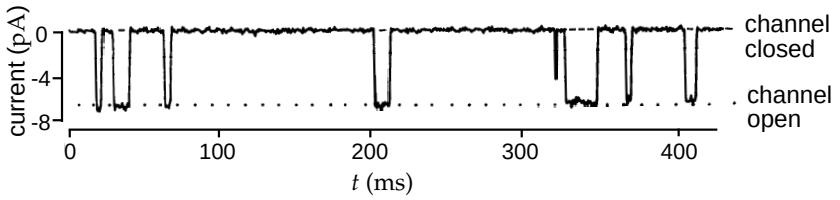

Figure 5.7 Recording of the current passing through a single ion channel. This is a synaptic receptor channel sensitive to the neurotransmitter acetylcholine. A small amount of acetylcholine was applied to the preparation to produce occasional channel openings. In the open state, the channel passes 6.6 pA at a holding potential of -140 mV. This is equivalent to more than 10^7 charges per second passing through the channel, and corresponds to an open channel conductance of 47 pS. (From Hille, 1992.)

associated with active membrane conductances. Recordings of the current flowing through single channels indicate that channels fluctuate rapidly between open and closed states in a stochastic manner (figure 5.7). Models of membrane and synaptic conductances must describe how the probability that a channel is in an open, ion-conducting state at any given time depends on the membrane potential (for a voltage-dependent conductance), the presence or absence of a neurotransmitter (for a synaptic conductance), or a number of other factors, such as the concentration of Ca^{2+} or other messenger molecules inside the cell. In this chapter, we consider two classes of active conductances, voltage-dependent membrane conductances and transmitter-dependent synaptic conductances. An additional type, the Ca^{2+}-dependent conductance, is considered in chapter 6.

stochastic channel

voltage-dependent, synaptic, and Ca^{2+}-dependent conductances

In a later section of this chapter, we discuss stochastic models of individual channels based on state diagrams and transition rates. However, most neuron models use deterministic descriptions of the conductances arising from many channels of a given type. This is justified because of the large number of channels of each type in the cell membrane of a typical neuron. If large numbers of channels are present, and if they fluctuate independently of each other (which they do, to a good approximation), then, from the law of large numbers, the fraction of channels open at any given time is approximately equal to the probability that any one channel is in an open state. This allows us to move between single-channel probabilistic formulations and macroscopic deterministic descriptions of membrane conductances.

We have denoted the conductance per unit area of membrane due to a set of ion channels of type i by g_i. The value of g_i at any given time is determined by multiplying the conductance of an open channel by the density of channels in the membrane and by the fraction of channels that are open at that time. The product of the first two factors is a constant called the maximal conductance that is denoted by $\overline{g}_i$. It is the conductance per unit area of membrane if all the channels of type i are open. Maximal conductance parameters tend to range from $\mu S/mm^2$ to mS/mm^2. The fraction of channels in the open state is equivalent to the probability of finding any

of inputs. In chapter 10, we study other models that construct output distributions in this way.

Note that the mean-field distribution $Q[\mathbf{v}]$ is simpler than the full Boltzmann distribution $P[\mathbf{v}]$ because the units are statistically independent. This prevents $Q[\mathbf{v}]$ from providing a good approximation in some cases, particularly if there are negative weights between units that tend to make their activities mutually exclusive. The mean-field analysis of the Boltzmann machine illustrates the limitations of rate-based descriptions in capturing the full extent of the correlations that can exist between spiking neurons.

7.7 Chapter Summary

The models in this chapter mark the start of our discussion of computation, as opposed to coding. Using a description of the firing rates of network neurons, we showed how to construct linear and nonlinear feedforward and recurrent networks that transform information from one coordinate system to another, selectively amplify input signals, integrate inputs over extended periods of time, select between competing inputs, sustain activity in the absence of input, exhibit gain modulation, allow simple decoding with performance near the Cramér-Rao bound, and act as content-addressable memories. We used network responses to a continuous stimulus variable as an extended example. This led to models of simple and complex cells in primary visual cortex. We described a model of the olfactory bulb as an example of a system for which computation involves oscillations arising from asymmetric couplings between excitatory and inhibitory neurons. Linear stability analysis was applied to a simplified version of this model. We also considered a stochastic network model called the Boltzmann machine.

7.8 Appendix

Lyapunov Function for the Boltzmann Machine

Here, we show that the Lyapunov function of equation 7.40 can be reduced to equation 7.59 when applied to the mean-field version of the Boltzmann machine. Recall, from equation 7.40, that

$$L(\mathbf{I}) = \sum_{a=1}^{N_v}\left(\int_0^{I_a} dz_a\, z_a F'(z_a) - h_a F(I_a) - \frac{1}{2}\sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'})\right). \quad (7.60)$$

When F is given by the sigmoidal function of equation 7.55,

$$\int_0^{I_a} dz_a\, z_a F'(z_a) = F(I_a)\ln F(I_a) + (1-F(I_a))\ln(1-F(I_a)) + k, \quad (7.61)$$

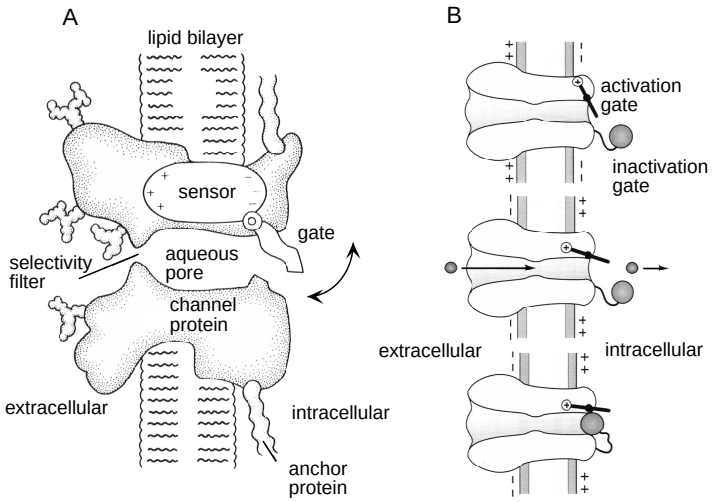

Figure 5.8 Gating of membrane channels. In both figures, the interior of the neuron is to the right of the membrane, and the extracellular medium is to the left. (A) A cartoon of gating of a persistent conductance. A gate is opened and closed by a sensor that responds to the membrane potential. The channel also has a region that selectively allows ions of a particular type to pass through the channel, for example, K^+ ions for a potassium channel. (B) A cartoon of the gating of a transient conductance. The activation gate is coupled to a voltage sensor (denoted by a circled +) and acts like the gate in A. A second gate, denoted by the ball, can block that channel once it is open. The top figure shows the channel in a deactivated (and deinactivated) state. The middle panel shows an activated channel, and the bottom panel shows an inactivated channel. Only the middle panel corresponds to an open, ion-conducting state. (A from Hille, 1992; B from Kandel et al., 1991.)

Although using the value of $k = 4$ is consistent with the four-subunit structure of the delayed-rectifier conductance, in practice k is an integer chosen to fit the data, and should be interpreted as a functional definition of a subunit rather than a reflection of a realistic structural model of the channel. Indeed, the structure of the channel was not known at the time that Hodgkin and Huxley chose the form of equation 5.15 and suggested that $k = 4$.

channel kinetics

We describe the transition of each subunit gate by a simple kinetic scheme in which the gating transition closed $\rightarrow$ open occurs at a voltage-dependent rate $\alpha_n(V)$, and the reverse transition, open $\rightarrow$ closed, occurs at a voltage-dependent rate $\beta_n(V)$. The probability that a subunit gate opens over a short interval of time is proportional to the probability of finding the gate closed, $1 - n$, multiplied by the opening rate $\alpha_n(V)$. Likewise, the probability that a subunit gate closes during a short time interval is proportional to the probability of finding the gate open, n, multiplied by the closing rate $\beta_n(V)$. The rate at which the open probability for a subunit gate changes is given by the difference of these two terms,

opening rate $\alpha_n(V)$

closing rate $\beta_n(V)$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n\,. \tag{5.16}$$

The state of unit a is determined by its total input current,

$$I_a(t) = h_a(t) + \sum_{a'=1}^{N_v} M_{aa'} v_{a'}(t)\,, \tag{7.54}$$

where $M_{aa'} = M_{a'a}$ and $M_{aa} = 0$ for all a and a', and h_a is the total feedforward input into unit a. In the model, units are permitted to change state only at integral multiples of Δt. At each time step, a single unit is selected, usually at random, to be updated. This update is based on a probabilistic rather than a deterministic rule. If unit a is selected, its state at the next time step is set stochastically to 1 with probability

$$P[v_a(t+\Delta t) = 1] = F(I_a(t))\,, \quad \text{with} \quad F(I_a) = \frac{1}{1+\exp(-I_a)}\,. \tag{7.55}$$

It follows that $P[v_a(t+\Delta t) = 0] = 1 - F(I_a(t))$. F is a sigmoidal function, which has the property that the larger the value of I_a, the more likely unit a is to take the value 1.

Markov chain

Under equation 7.55, the state of activity of the network evolves as a Markov chain. This means that the components of $\mathbf{v}$ at different times are sequences of random variables with the property that $\mathbf{v}(t+\Delta t)$ depends only on $\mathbf{v}(t)$, and not on the previous history of the network. Equation 7.55 implements what is known as Glauber dynamics.

Glauber dynamics

An advantage of using Glauber dynamics to define the evolution of a network model is that general results from statistical mechanics can be used to determine the equilibrium distribution of activities. Under Glauber dynamics, $\mathbf{v}$ does not converge to a fixed point, but can be described by a probability distribution associated with an energy function

energy function

$$E(\mathbf{v}) = -\mathbf{h}\cdot\mathbf{v} - \frac{1}{2}\mathbf{v}\cdot\mathbf{M}\cdot\mathbf{v}\,. \tag{7.56}$$

The probability distribution characterizing $\mathbf{v}$, once the network has converged to an equilibrium state, is

$$P[\mathbf{v}] = \frac{\exp(-E(\mathbf{v}))}{Z} \quad \text{where} \quad Z = \sum_{\mathbf{v}} \exp(-E(\mathbf{v}))\,. \tag{7.57}$$

The notion of convergence as $t \to \infty$ can be made precise; but informally it means that after repeated updating according to equation 7.55, the states of the network are described statistically by equation 7.57. Z is called the partition function and $P[\mathbf{v}]$, the Boltzmann distribution. Under the Boltzmann distribution, states with lower energies are more likely. In this case, Glauber dynamics implements a statistical operation called Gibbs sampling for the distribution given in equation 7.57. From now on, we refer to the update procedure described by equation 7.55 as Gibbs sampling.

partition function

Boltzmann distribution

Gibbs sampling

mean-field approximation

The Boltzmann machine is inherently stochastic. However, an approximation to the Boltzmann machine, known as the mean-field approximation,

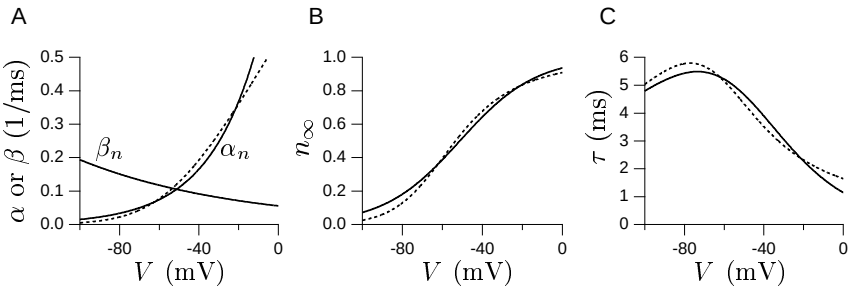

Figure 5.9 Generic voltage-dependent gating functions compared with Hodgkin-Huxley results for the delayed-rectifier K^+ conductance. (A) The exponential α_n and β_n functions expected from thermodynamic arguments are indicated by the solid curves. Parameter values used were $A_\alpha = 1.22$ ms^{-1}, $A_\beta = 0.056$ ms^{-1}, $B_\alpha/V_T = -0.04$/mV, and $B_\beta/V_T = 0.0125$/mV. The fit of Hodgkin and Huxley for β_n is identical to the solid curve shown. The Hodgkin-Huxley fit for α_n is the dashed curve. (B) The corresponding function $n_\infty(V)$ of equation 5.21 (solid curve). The dashed curve is obtained using the α_n and β_n functions of the Hodgkin-Huxley fit (equation 5.22). (C) The corresponding function $\tau_n(V)$, obtained from equation 5.18 (solid curve). Again the dashed curve is the result of using the Hodgkin-Huxley rate functions.

While thermodynamic arguments support the forms we have presented, they rely on simplistic assumptions. Not surprisingly, the resulting functional forms do not always fit the data, and various alternatives are often employed. The data upon which these fits are based are typically obtained using a technique called voltage clamping. In this technique, an amplifier is configured to inject the appropriate amount of electrode current to hold the membrane potential at a constant value. By current conservation, this current is equal to the membrane current of the cell. Hodgkin and Huxley fitted the rate functions for the delayed-rectifier K^+ conductance they studied, using the equations

voltage clamping

$$\alpha_n = \frac{.01(V+55)}{1-\exp(-.1(V+55))} \quad \text{and} \quad \beta_n = 0.125\exp(-0.0125(V+65))\,, \tag{5.22}$$

where V is expressed in mV, and α_n and β_n are both expressed in units of 1/ms. The fit for β_n is exactly the exponential form we have discussed, with $A_\beta = 0.125\exp(-0.0125\cdot 65)$ ms^{-1} and $B_\beta/V_T = 0.0125$ mV^{-1}, but the fit for α_n uses a different functional form. The dashed curves in figure 5.9 plot the formulas of equation 5.22.

Transient Conductances

Some channels only open transiently when the membrane potential is depolarized because they are gated by two processes with opposite voltage dependences. Figure 5.8B is a schematic of a channel that is controlled by two gates and generates a transient conductance. The swinging gate in figure 5.8B behaves exactly like the gate in figure 5.8A. The probability that it

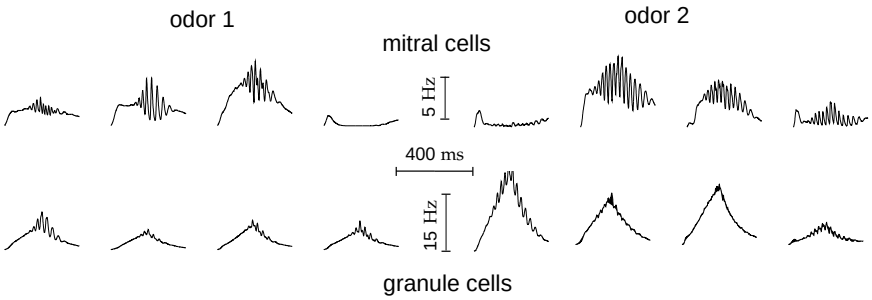

Figure 7.22 Activities of four of ten mitral (upper) and granule (lower) cells during a single sniff cycle for two different odors. (Adapted from Li and Hopfield, 1989.)

modifying where the fixed point lies on the activation function curves in figure 7.21A. Second, it affects which particular neurons are destabilized, and thus which begin to oscillate. The ultimate pattern of oscillatory activity is determined both by the input $\mathbf{h}_E$ and by the recurrent couplings of the network.

Figure 7.22 shows the behavior of the network during a single sniff cycle in the presence of two different odors, represented by two different values of $\mathbf{h}_E$. The top rows show the activity of four mitral cells, and the bottom rows of four granule cells. The amplitudes and phases of the oscillations seen in these traces, along with the identities of the mitral cells taking part in them, provide a signature of the identity of the odor that was presented.

Oscillatory Amplification

As a final example of network oscillations, we return to amplification of input signals by a recurrently connected network. Two factors control the amount of selective amplification that is viable in networks such as that shown in figure 7.9. The most important constraint on the recurrent weights is that the network must be stable, so the activity does not increase without bound. Another possible constraint is suggested by figure 7.14D, where the output shows a tuned response even though the input to the network is constant as a function of θ. Tuned output in the absence of tuned input can serve as a memory mechanism, but it will produce persistent perceptions if it occurs in a primary sensory area, for example. Avoiding this in the network limits the recurrent weights and the amount of amplification that can be supported.

Li and Dayan (1999) showed that this restriction can be significantly eased using the richer dynamics of networks of coupled inhibitory and excitatory neurons. Figure 7.23 shows an example with continuous neuron labeling based on a continuous version of equations 7.12 and 7.13. The input is $h_E(\theta) = 8 + 5\cos(2\theta)$ in the modulated case (figure 7.23B) or $h_E(\theta) = 8$ in the unmodulated case (figure 7.23C). Noise with standard deviation 0.4 corrupts this input. The input to the network is constant in time.

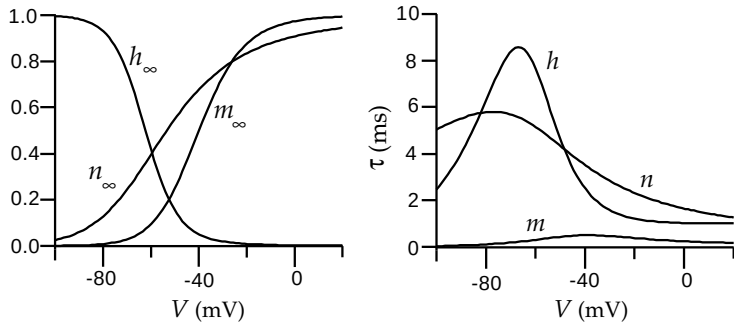

Figure 5.10 The voltage-dependent functions of the Hodgkin-Huxley model. The left panel shows $m_\infty(V)$, $h_\infty(V)$, and $n_\infty(V)$, the steady-state levels of activation and inactivation of the Na^+ conductance, and activation of the K^+ conductance. The right panel shows the voltage-dependent time constants that control the rates at which these steady-state levels are approached for the three gating variables.

gate. Another class of conductances, the hyperpolarization-activated conductances, behave as if they are controlled solely by an inactivation gate. They are thus persistent conductances, but they open when the neuron is hyperpolarized rather than depolarized. The opening probability for such channels is written solely in terms of an inactivation variable similar to h. Strictly speaking, these conductances deinactivate when they turn on and inactivate when they turn off. However, most people cannot bring themselves to say "deinactivate" all the time, so they say instead that these conductances are activated by hyperpolarization.

5.6 The Hodgkin-Huxley Model

The Hodgkin-Huxley model for the generation of the action potential, in its single-compartment form, is constructed by writing the membrane current in equation 5.6 as the sum of a leakage current, a delayed-rectified K^+ current, and a transient Na^+ current,

$$i_m = \overline{g}_L(V - E_L) + \overline{g}_K n^4(V - E_K) + \overline{g}_{Na} m^3 h(V - E_{Na}) . \qquad (5.25)$$

The maximal conductances and reversal potentials used in the model are $\overline{g}_L$ = 0.003 mS/mm^2, $\overline{g}_K$ = 0.36 mS/mm^2, $\overline{g}_{Na}$ = 1.2 mS/mm^2, E_L = -54.387 mV, E_K = -77 mV and E_{Na} = 50 mV. The full model consists of equation 5.6 with equation 5.25 for the membrane current, and equations of the form 5.17 for the gating variables n, m, and h. These equations can be integrated numerically, using the methods described in appendices A and B.

The temporal evolution of the dynamic variables of the Hodgkin-Huxley model during a single action potential is shown in figure 5.11. The initial rise of the membrane potential, prior to the action potential, seen in the upper panel of figure 5.11, is due to the injection of a positive electrode current into the model starting at $t = 5$ ms. When this current drives the

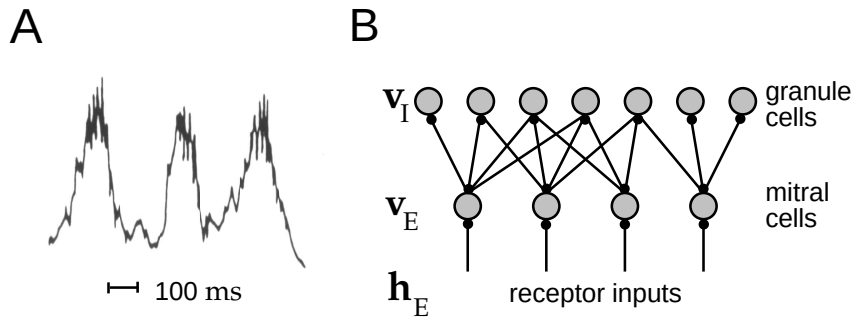

Figure 7.20 (A) Extracellular field potential recorded in the olfactory bulb during respiratory waves representing three successive sniffs. (B) Schematic diagram of the olfactory bulb model. (A adapted from Freeman and Schneider, 1982; B adapted from Li, 1995.)

The Olfactory Bulb

mitral cells
tufted cells
granule cells

The olfactory bulb, and analogous olfactory areas in insects, provide examples of sensory processing involving oscillatory activity. The olfactory bulb represents the first stage of processing beyond the olfactory receptors in the vertebrate olfactory system. Olfactory receptor neurons respond to odor molecules and send their axons to the olfactory bulb. These axons terminate in glomeruli where they synapse onto mitral and tufted cells, as well as local interneurons. The mitral and tufted cells provide the output of the olfactory bulb by sending projections to the primary olfactory cortex. They also synapse onto the larger population of inhibitory granule cells. The granule cells in turn inhibit the mitral and tufted cells.

The activity in the olfactory bulb of many vertebrates is strongly influenced by a sniff cycle in which a few quick sniffs bring odors past the olfactory receptors. Figure 7.20A shows an extracellular potential recorded during three successive sniffs. The three large oscillations in the figure are due to the sniffs. The oscillations we discuss in this section are the smaller, higher-frequency oscillations seen around the peak of each sniff cycle. These arise from oscillatory neural activity. Individual mitral cells have quite low firing rates, and do not fire on each cycle of the oscillations. The oscillations are phase-locked across the bulb, in that different neurons fire at fixed phase lags from each other, but different odors induce oscillations of different amplitudes and phases.

Li and Hopfield (1989) modeled the mitral and granule cells of the olfactory bulb as a nonlinear input-driven network oscillator. Figure 7.20B shows the architecture of the model, which uses equations 7.12 and 7.13 with $\mathbf{M}_{EE} = \mathbf{M}_{II} = 0$. The absence of these couplings in the model is in accord with the anatomy of the bulb. The rates $\mathbf{v}_E$ and $\mathbf{v}_I$ refer to the mitral and granule cells, respectively. Figure 7.21A shows the activation functions of the model. The time constants for the two populations of cells are the same, $\tau_E = \tau_I = 6.7$ ms. $\mathbf{h}_E$ is the input from the receptors to the mitral

The Hodgkin-Huxley model can also be used to study propagation of an action potential down an axon, but for this purpose a multi-compartment model must be constructed. Methods for building such a model, and results from it, are described in chapter 6.

5.7 Modeling Channels

In previous sections, we described the Hodgkin-Huxley formalism for describing voltage-dependent conductances arising from a large number of channels. With the advent of single-channel studies, microscopic descriptions of the transitions between the conformational states of channel molecules have been developed. Because these models describe complex molecules, they typically involve many states and transitions. Here, we discuss simple versions of these models that capture the spirit of single-channel modeling without getting mired in the details.

Models of single channels are based on state diagrams that indicate the possible conformational states that the channel can assume. Typically, one of the states in the diagram is designated as open and ion-conducting, while the other states are nonconducting. The current conducted by the channel is written as $\overline{g}P(V - E)$, where E is the reversal potential, $\overline{g}$ is the single-channel open conductance, and P is 1 whenever the open state is occupied, and 0 otherwise. Channel models can be instantiated directly from state diagrams simply by keeping track of the state of the channel and allowing stochastic changes of state to occur at appropriate transition rates. If the model is updated in short time steps of duration Δt, the probability that the channel makes a given transition during an update interval is the transition rate times Δt.

Figure 5.12 shows the state diagram and simulation results for a model of a single delayed-rectifier K^+ channel that is closely related to the Hodgkin-Huxley description of the macroscopic delayed-rectifier conductance. The factors α_n and β_n in the transition rates shown in the state diagram of figure 5.12 are the voltage-dependent rate functions of the Hodgkin-Huxley model. The model uses the same four subunit structure assumed in the Hodgkin-Huxley model. We can think of state 1 in this diagram as a state in which all the subunit gates are closed. States 2, 3, 4, and 5 have 1, 2, 3, and 4 open subunit gates, respectively. State 5 is the sole open state. The factors of 1, 2, 3, and 4 in the transition rates in figure 5.12 correspond to the number of subunit gates that can make a given transition. For example, the transition rate from state 1 to state 2 is four times faster than the rate from state 4 to state 5. This is because any one of the four subunit gates can open to get from state 1 to state 2, but the transition from state 4 to state 5 requires the single remaining closed subunit gate to open.

The lower panels in figure 5.12 show simulations of this model involving 1, 10, and 100 channels. The sum of currents from all of these channels is compared with the current predicted by the Hodgkin-Huxley model

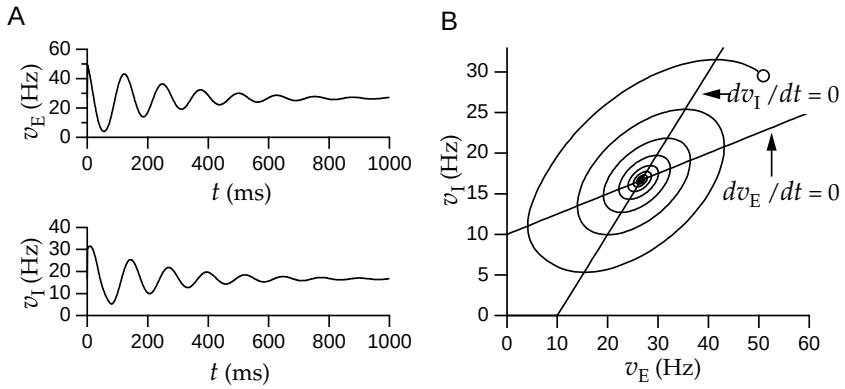

Figure 7.18 Activity of the excitatory-inhibitory firing-rate model when the fixed point is stable. (A) The excitatory and inhibitory firing rates settle to the fixed point over time. (B) The phase-plane trajectory is a counterclockwise spiral collapsing to the fixed point. The open circle marks the initial values $v_E(0)$ and $v_I(0)$. For this example, $\tau_I = 30$ ms.

and 7.51 by τ_E and τ_I respectively. We then evaluate these derivatives at the values of v_E and v_I that correspond to the fixed point. The four combinations of derivatives computed in this way can be arranged into a matrix

stability matrix

$$\begin{pmatrix} (M_{EE}-1)/\tau_E & M_{EI}/\tau_E \\ M_{IE}/\tau_I & (M_{II}-1)/\tau_I \end{pmatrix}. \tag{7.52}$$

As discussed in the Mathematical Appendix, the stability of the fixed point is determined by the real parts of the eigenvalues of this matrix. The eigenvalues are given by

$$\lambda = \frac{1}{2}\left(\frac{M_{EE}-1}{\tau_E} + \frac{M_{II}-1}{\tau_I} \pm \sqrt{\left(\frac{M_{EE}-1}{\tau_E} - \frac{M_{II}-1}{\tau_I}\right)^2 + \frac{4M_{EI}M_{IE}}{\tau_E\tau_I}}\right). \tag{7.53}$$

If the real parts of both eigenvalues are less than 0, the fixed point is stable, whereas if either is greater than 0, the fixed point is unstable. If the factor under the radical sign in equation 7.53 is positive, both eigenvalues are real, and the behavior near the fixed point is exponential. This means that there is exponential movement toward the fixed point if both eigenvalues are negative, or away from the fixed point if either eigenvalue is positive. We focus on the case when the factor under the radical sign is negative, so that the square root is imaginary and the eigenvalues form a complex conjugate pair. In this case, the behavior near the fixed point is oscillatory and the trajectory either spirals into the fixed point, if the real part of the eigenvalues is negative, or out from the fixed point if the real part of the eigenvalues is positive. The imaginary part of the eigenvalue determines the frequency of oscillations near the fixed point. The real and imaginary parts of one of these eigenvalues are plotted as a function of τ_I

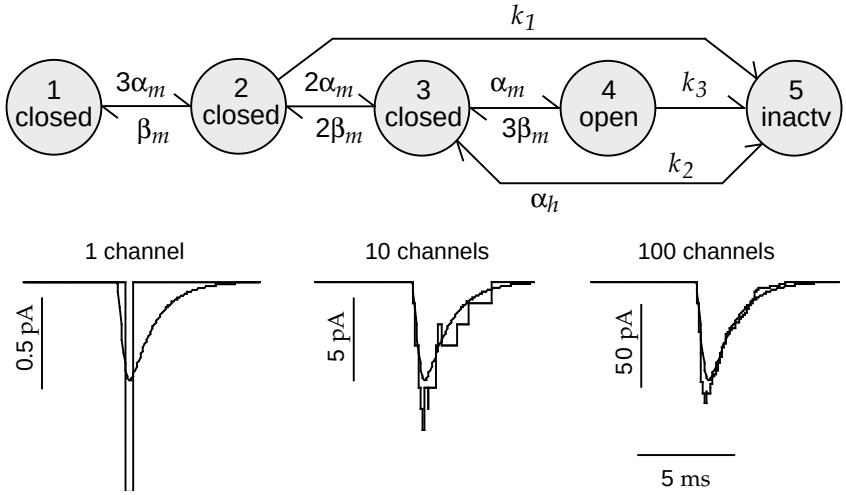

Figure 5.13 A model of the fast Na^+ channel. The upper diagram shows the states and transitions rates of the model. The values $k_1 = 0.24/\text{ms}$, $k_2 = 0.4/\text{ms}$, and $k_3 = 1.5/\text{ms}$ were used in the simulations shown in the lower panels. For these simulations, the membrane potential was initially held at -100 mV, then held at 10 mV for 20 ms, and finally returned to a holding potential of -100 mV. The smooth curves in these panels show the current predicted by the Hodgkin-Huxley model in this situation. The left panel shows a simulation of a single channel that opened once during the depolarization. The middle panel shows the total current from 10 simulated channels, and the right panel corresponds to 100 channels. As the number of channels increases, the Hodgkin-Huxley model provides a fairly accurate description of the current, but it is not identical to the channel model in this case.

Hodgkin-Huxley model, n is the probability of a subunit gate being in the open state and $1 - n$ is the probability of it being closed. If we use that same notation here, state 1 has four closed subunit gates, and thus $p_1 = (1 - n)^4$. State 5, the open state, has four open subunit gates, so $p_5 = n^4 = P$. State 2 has one open subunit gate, which can be any one of the four subunit gates, and three closed states, making $p_2 = 4n(1 - n)^3$. Similar arguments yield $p_3 = 6n^2(1 - n)^2$ and $p_4 = 4n^3(1 - n)$. These expressions generate a solution to the above equations, provided that n satisfies equation 5.16, as the reader can verify.

In the Hodgkin-Huxley model of the Na^+ conductance, the activation and inactivation processes are assumed to act independently. The schematic in figure 5.8B, which cartoons the mechanism believed to be responsible for inactivation, suggests that this assumption is incorrect. The ball that inactivates the channel is located inside the cell membrane, where it cannot be affected directly by the potential across the membrane. Furthermore, in this scheme the ball cannot occupy the channel pore until the activation gate has opened, making the two processes interdependent.

The state diagram in figure 5.13 reflects this by having a state-dependent, voltage-independent inactivation mechanism. This diagram is a simplified version of an Na^+ channel model due to Patlak (1991). The sequence of transitions that lead to channel opening through states 1, 2, 3, and 4 is

state-dependent inactivation

Homogeneous Excitatory and Inhibitory Populations

As an illustration of the dynamics of excitatory-inhibitory network models, we analyze a simple model in which all of the excitatory neurons are described by a single firing rate, v_E, and all of the inhibitory neurons are described by a second rate, v_I. Although we think of this example as a model of interacting neuronal populations, it is constructed as if it consists of just two neurons. Equations 7.12 and 7.13, with threshold linear response functions, are used to describe the two firing rates, so that

$$\tau_E \frac{dv_E}{dt} = -v_E + [M_{EE}v_E + M_{EI}v_I - \gamma_E]_+ \tag{7.50}$$

and

$$\tau_I \frac{dv_I}{dt} = -v_I + [M_{II}v_I + M_{IE}v_E - \gamma_I]_+ \ . \tag{7.51}$$

The synaptic weights M_{EE}, M_{IE}, M_{EI}, and M_{II} are numbers rather than matrices in this model. In the example we consider, we set $M_{EE} = 1.25$, $M_{IE} = 1$, $M_{II} = 0$, $M_{EI} = -1$, $\gamma_E = -10$ Hz, $\gamma_I = 10$ Hz, $\tau_E = 10$ ms, and we vary the value of τ_I. The negative value of γ_E means that this parameter serves as a source of constant background activity rather than as a threshold.

Phase-Plane Methods and Stability Analysis

The model of interacting excitatory and inhibitory populations given by equations 7.50 and 7.51 provides an opportunity for us to illustrate some of the techniques used to study the dynamics of nonlinear systems. This model exhibits both fixed-point (constant v_E and v_I) and oscillatory activity, depending on the values of its parameters. Stability analysis can be used to determine the parameter values where transitions between these two types of activity take place.

The firing rates $v_E(t)$ and $v_I(t)$ arising from equations 7.50 and 7.51 can be displayed by plotting them as functions of time, as in figures 7.18A and 7.19A. Another useful way of depicting these results, illustrated in figures 7.18B and 7.19B, is to plot pairs of points $(v_E(t), v_I(t))$ for a range of t values. As the firing rates change, these points trace out a curve or trajectory in the v_E-v_I plane, which is called the phase plane of the model. Phase-plane plots can be used to give a geometric picture of the dynamics of a model.

phase plane

Values of v_E and v_I for which the right side of either equation 7.50 or equation 7.51 vanishes are of particular interest in phase-plane analysis. Sets of such values form two curves in the phase plane known as nullclines. The nullclines for equations 7.50 and 7.51 are the straight lines drawn in figure 7.17A. The nullclines are important because they divide the phase plane into regions with opposite flow patterns. This is because dv_E/dt and

nullcline

the pre- and postsynaptic sides of the synapse, $P = P_s P_{rel}$. The factor P_s is the probability that a postsynaptic channel opens, given that the transmitter was released by the presynaptic terminal. Because there are typically many postsynaptic channels, this can also be taken as the fraction of channels opened by the transmitter.

synaptic open probability P_s

P_{rel} is related to the probability that transmitter is released by the presynaptic terminal following the arrival of an action potential. This reflects the fact that transmitter release is a stochastic process. Release of transmitter at a presynaptic terminal does not necessarily occur every time an action potential arrives and, conversely, spontaneous release can occur even in the absence of the depolarization due to an action potential. The interpretation of P_{rel} is a bit subtle because a synaptic connection between neurons may involve multiple anatomical synapses, and each of these may have multiple independent transmitter release sites. The factor P_{rel}, in our discussion, is the average of the release probabilities at each release site. If there are many release sites, the total amount of transmitter released by all the sites is proportional to P_{rel}. If there is a single release site, P_{rel} is the probability that it releases transmitter. We will restrict our discussion to these two interpretations of P_{rel}. For a modest number of release sites with widely varying release probabilities, the current we discuss describes only an average over multiple trials.

transmitter release probability P_{rel}

Synapses can exert their effects on the soma, dendrites, axon spike-initiation zone, or presynaptic terminals of their postsynaptic targets. There are two broad classes of synaptic conductances that are distinguished by whether the transmitter binds to the synaptic channel and activates it directly, or the transmitter binds to a distinct receptor that activates the conductance indirectly through an intracellular signaling pathway. The first class is called ionotropic and the second, metabotropic. Ionotropic conductances activate and deactivate more rapidly than metabotropic conductances. Metabotropic receptors can, in addition to opening channels, cause long-lasting changes inside a neuron. They typically operate through pathways that involve G-protein-mediated receptors and various intracellular signaling molecules known as second messengers. Many neuromodulators, including serotonin, dopamine, norepinephrine, and acetylcholine, act through metabotropic receptors. These have a wide variety of important effects on the functioning of the nervous system.

ionotropic receptor

metabotropic receptor

Glutamate and GABA (γ-aminobutyric acid) are the major excitatory and inhibitory transmitters in the brain. Both act ionotropically and metabotropically. The principal ionotropic receptor types for glutamate are called AMPA and NMDA. Both AMPA and NMDA receptors produce mixed-cation conductances with reversal potentials around 0 mV. The AMPA current activates and deactivates rapidly. The NMDA receptor is somewhat slower to activate and deactivates considerably more slowly. In addition, NMDA receptors have an unusual voltage dependence that we discuss in a later section, and are more permeable to Ca^{2+} than AMPA receptors.

glutamate, GABA

AMPA, NMDA

The reader is urged to verify that, due to the additional terms in the sum over memory patterns, the conditions that must be satisfied when using 7.48 are slightly modified from 7.46 to

$$F(-c(1+\alpha\lambda)) = 0 \quad \text{and} \quad c = F(c(\lambda - 1 - \alpha\lambda)) \,. \tag{7.49}$$

One way of looking at the recurrent weights in equation 7.48 is in terms of a learning rule used to construct the matrix. In this learning rule, an excitatory contribution to the coupling between two units is added whenever both of them are either active or inactive for a particular memory pattern. An inhibitory term is added whenever one unit is active and the other is not. The learning rule associated with equation 7.48 is called a covariance rule because of its relationship to the covariance matrix of the memory patterns. Learning rules for constructing networks that perform associative memory and other tasks are discussed in chapter 8.

Figure 7.16 shows an associative memory network of $N_v = 50$ units that stores four patterns, using the matrix from equation 7.48. Two of these patterns were generated randomly as discussed above. The other two patterns were assigned nonrandomly to make them easy to identify in the figure. Recall of these two nonrandom patterns is shown in figures 7.16B and 7.16C. From an initial pattern of activity only vaguely resembling one of the stored patterns, the network attains a fixed point very similar to the best matching memory pattern. The same results apply for the other two memory patterns stored by the network, but they are more difficult to identify in a figure because they are random.

The matrix 7.48 that we use as a basis for constructing an associative memory network satisfies the conditions required for exact storage and recall of the memory patterns only approximately. This introduces some errors in recall. As the number of memory patterns increases, the approximation becomes worse and the performance of the associative memory deteriorates, which limits the number of memories that can be stored. The simple covariance prescription for the weights in equation 7.48 is far from optimal. Other prescriptions for constructing $\mathbf{M}$ can achieve significantly higher storage capacities.

The basic conclusions from studies of associative memory models is that large networks can store large numbers of patterns, particularly if they are sparse (α is small) and if a few errors in recall can be tolerated. The capacity of certain associative memory networks can be calculated analytically. The number of memory patterns that can be stored is on the order of, but typically less than, the number of neurons in the network, N_v, and depends on the sparseness, α, as $1/(\alpha \log(1/\alpha))$. The amount of information that can be stored is proportional to N_v^2, which is roughly the number of synapses in the network, but the information stored per synapse (i.e., the constant of proportionality) is typically quite small.

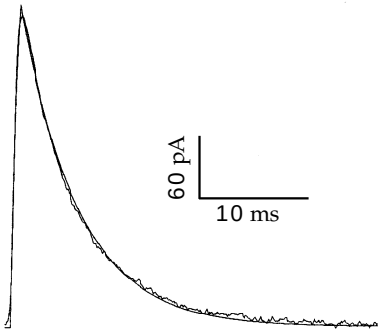

Figure 5.14 A fit of the model discussed in the text to the average EPSC (excitatory postsynaptic current) recorded from mossy fiber input to a CA3 pyramidal cell in a hippocampal slice preparation. The smooth line is the theoretical curve and the wiggly line is the result of averaging recordings from a number of trials. (Adapted from Destexhe et al., 1994.)

is nonzero, α_s is so much larger than β_s that we can ignore the term involving β_s in equation 5.27. Integrating equation 5.27 under this assumption, we find that

$$P_s(t) = 1 + (P_s(0) - 1)\exp(-\alpha_s t) \quad \text{for} \quad 0 \le t \le T\,. \tag{5.28}$$

The open probability takes its maximum value at time $t = T$ and then, for $t \ge T$, decays exponentially at a rate determined by the constant β_s,

$$P_s(t) = P_s(T)\exp(-\beta_s(t - T)) \quad \text{for} \quad t \ge T\,. \tag{5.29}$$

If $P_s(0) = 0$, as it will if there is no synaptic release immediately before the release at $t = 0$, equation 5.28 simplifies to $P_s(t) = 1 - \exp(-\alpha_s t)$ for $0 \le t \le T$, and this reaches a maximum value $P_{max} = P_s(T) = 1 - \exp(-\alpha_s T)$. In terms of this parameter, a simple manipulation of equation 5.28 shows that we can write, in the general case,

$$P_s(T) = P_s(0) + P_{max}(1 - P_s(0))\,. \tag{5.30}$$

Figure 5.14 shows a fit to a recorded postsynaptic current using this formalism. In this case, β_s was set to 0.19 ms^{-1}. The transmitter concentration was modeled as a square pulse of duration $T = 1$ ms during which $\alpha_s = 0.93$ ms^{-1}. Inverting these values, we find that the time constant determining the rapid rise seen in figure 5.14A is 0.9 ms, while the fall of the current is an exponential with a time constant of 5.26 ms.

For a fast synapse like the one shown in figure 5.14, the rise of the conductance following a presynaptic action potential is so rapid that it can be approximated as instantaneous. In this case, the synaptic conductance due to a single presynaptic action potential occurring at $t = 0$ is often written as an exponential, $P_s = P_{max}\exp(-t/\tau_s)$ (see the AMPA trace in figure 5.15A), where from equation 5.29, $\tau_s = 1/\beta_s$. The synaptic conductance due to a

equal" means that a significant number, but not necessarily all, of the elements of $\mathbf{v}(0)$ are close to the corresponding elements of $\mathbf{v}^m$. The network then evolves according to equation 7.11. If recall is successful, the dynamics converge to a fixed point equal (or at least significantly more similar than $\mathbf{v}(0)$) to the memory pattern associated with the initial state (i.e., $\mathbf{v}(t) \to \mathbf{v}^m$ for large t). Failure of recall occurs if the fixed point reached by the network is not similar to $\mathbf{v}^m$, or if a fixed point is not reached at all.

For exact recall to occur, $\mathbf{v}^m$ must be a fixed point of the network dynamics, which means it must satisfy the equation

$$\mathbf{v}^m = \mathbf{F}(\mathbf{M} \cdot \mathbf{v}^m) \,. \tag{7.42}$$

memory capacity

Therefore, we examine conditions under which such solutions exist for all the memory patterns. The capacity of a network is determined in part by the number of different pre-specified vectors that can simultaneously satisfy equation 7.42 for an appropriate choice of $\mathbf{M}$. In the limit of large N_v, the capacity is typically proportional to N_v. Capacity is not the only relevant measure of the performance of an associative memory. Memory function can be degraded if there are spurious fixed points of the network dynamics in addition to the fixed points that represent the memory patterns. Finally, useful pattern matching requires each fixed point to have a sufficiently large basin of attraction. Analyzing spurious fixed points and the sizes of basins of attraction is beyond the scope of this text.

Although the units in the network have continuous-valued activities, we consider the simple case in which the units are either inactive or active in the memory patterns themselves. Inactive units correspond to components of $\mathbf{v}^m$ that are equal to 0, and active units, to components that are equal to some constant value c. To simplify the discussion, we assume that each of the memory patterns has exactly αN_v active and $(1-\alpha)N_v$ inactive units. The choice of which units are active in each pattern is random, and independent of the other patterns. The parameter α is known as the sparseness of the memory patterns. As α decreases, making the patterns more sparse, more of them can be stored but each contains less information.

sparseness α

To build an associative memory network, we need to construct a matrix that allows all the memory patterns to satisfy equation 7.42. To begin, suppose that we knew of a matrix $\mathbf{K}$ for which all the memory patterns were degenerate eigenvectors with eigenvalue λ,

$$\mathbf{K} \cdot \mathbf{v}^m = \lambda \mathbf{v}^m \tag{7.43}$$

for all m. Then, consider the matrix

$$\mathbf{M} = \mathbf{K} - \frac{\mathbf{nn}}{\alpha N_v} \quad \text{or} \quad M_{aa'} = K_{aa'} - \frac{1}{\alpha N_v} \,. \tag{7.44}$$

vector of ones $\mathbf{n}$

Here $\mathbf{n}$ is a vector that has each of its N_v components equal to 1. The term $\mathbf{nn}$ in the matrix represents uniform inhibition between network units. $\mathbf{M}$ satisfies

$$\mathbf{M} \cdot \mathbf{v}^m = \lambda \mathbf{v}^m - c\mathbf{n} \tag{7.45}$$

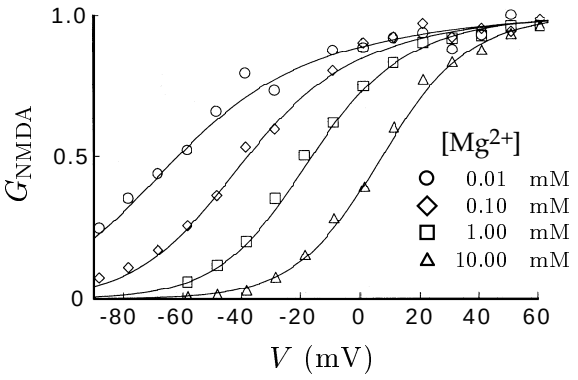

Figure 5.16 Dependence of the NMDA conductance on the membrane potential and extracellular Mg^{2+} concentration. Normal extracellular Mg^{2+} concentrations are in the range of 1 to 2 mM. The solid lines are the factors G_{NMDA} of equation 5.36 for different values of $[Mg^{2+}]$, and the symbols indicate the data points. (Adapted from Jahr and Stevens, 1990.)

for an isolated presynaptic release that occurs at time $t = 0$. This expression, called an alpha function, starts at 0, reaches its peak value at $t = \tau_s$, and then decays with a time constant τ_s. *alpha function*

We mentioned earlier in this chapter that NMDA receptor conductance has an additional dependence on the postsynaptic potential not normally seen in other conductances. To incorporate this dependence, the current due to the NMDA receptor can be described using an additional factor that depends on the postsynaptic potential, V. The NMDA current is written as $\overline{g}_{\mathrm{NMDA}} G_{\mathrm{NMDA}}(V) P(V - E_{\mathrm{NMDA}})$. P is the usual open probability factor. The factor $G_{\mathrm{NMDA}}(V)$ describes an extra voltage dependence due to the fact that when the postsynaptic neuron is near its resting potential, NMDA receptors are blocked by Mg^{2+} ions. To activate the conductance, the postsynaptic neuron must be depolarized to knock out the blocking ions. Jahr and Stevens (1990) have fitted this dependence by (figure 5.16) *NMDA receptor*

$$G_{\mathrm{NMDA}} = \left(1 + \frac{[Mg^{2+}]}{3.57\ \mathrm{mM}} \exp(-V/16.13\ \mathrm{mV})\right)^{-1}. \qquad (5.36)$$

NMDA receptors conduct Ca^{2+} ions as well as monovalent cations. Entry of Ca^{2+} ions through NMDA receptors is a critical event for long-term modification of synaptic strength. The fact that the opening of NMDA channels requires both pre- and postsynaptic depolarization means NMDA receptors can act as coincidence detectors of simultaneous pre- and postsynaptic activity. This plays an important role in connection with the Hebb rule for synaptic modification discussed in chapter 8. *coincidence detection*

Network Stability

fixed-point

When a network responds to a constant input by relaxing to a steady state with $d\mathbf{v}/dt = \mathbf{0}$, it is said to exhibit fixed-point behavior. Almost all the network activity we have discussed thus far involves such fixed points. This is by no means the only type of long-term activity that a network model can display. In a later section of this chapter, we discuss networks that oscillate, and chaotic behavior is also possible. But if certain conditions are met, a network will inevitably reach a fixed point in response to constant input. The theory of Lyapunov functions, to which we give an informal introduction, can be used to prove when this occurs.

It is easier to discuss the Lyapunov function for a network if we use the firing-rate dynamics of equation 7.6 rather than equation 7.8. For a network model, this means expressing the vector of network firing rates as $\mathbf{v} = \mathbf{F}(\mathbf{I})$, where $\mathbf{I}$ is the total synaptic current vector (i.e., I_a represents the total synaptic current for unit a). We assume that $F'(I) > 0$ for all I, where F' is the derivative of F. $\mathbf{I}$ obeys the dynamic equation derived from generalizing equation 7.6 to a network situation,

recurrent model with current dynamics

$$\tau_s \frac{d\mathbf{I}}{dt} = -\mathbf{I} + \mathbf{h} + \mathbf{M} \cdot \mathbf{F}(\mathbf{I}) \, . \tag{7.39}$$

Note that we have made the substitution $\mathbf{v} = \mathbf{F}(\mathbf{I})$ in the last term of the right side of this equation. Equation 7.39 can be used instead of equation 7.11 to provide a firing-rate model of a recurrent network.

Lyapunov function L

For the firing-rate model of equation 7.39 with a symmetric recurrent weight matrix, Cohen and Grossberg (1983) showed that the function

$$L(\mathbf{I}) = \sum_{a=1}^{N_v} \left(\int_0^{I_a} dz_a \, z_a F'(z_a) - h_a F(I_a) - \frac{1}{2} \sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'}) \right) \tag{7.40}$$

has $dL/dt < 0$ whenever $d\mathbf{I}/dt \neq \mathbf{0}$. To see this, take the time derivative of equation 7.40 and use 7.39 to obtain

$$\frac{dL(\mathbf{I})}{dt} = -\tau_s \sum_{a=1}^{N_v} F'(I_a) \left(\frac{dI_a}{dt} \right)^2 \, . \tag{7.41}$$

Because $F' > 0$, L decreases unless $d\mathbf{I}/dt = \mathbf{0}$. If L is bounded from below, it cannot decrease indefinitely, so $\mathbf{I} = \mathbf{h} + \mathbf{M} \cdot \mathbf{v}$ must converge to a fixed point. This implies that $\mathbf{v}$ must converge to a fixed point as well.

We have required that $F'(I) > 0$ for all values of its argument I. However, with some technical complications, it can be shown that the Lyapunov function we have presented also applies to the case of the rectifying activation function $F(I) = [I]_+$, even though it is not differentiable at $I = 0$ and $F'(I) = 0$ for $I < 0$. Convergence to a fixed point, or one of a set of fixed points, requires the Lyapunov function to be bounded from below. One way to ensure this is to use a saturating activation function, so that $F(I)$ is bounded as $I \to \infty$. Another way is to keep the eigenvalues of $\mathbf{M}$ sufficiently small.

Facilitation and depression can both be modeled as presynaptic processes that modify the probability of transmitter release. We describe them using a simple nonmechanistic model that has similarities to the model of P_s presented in the previous subsection. For both facilitation and depression, the release probability after a long period of presynaptic silence is $P_{\rm rel} = P_0$. Activity at the synapse causes $P_{\rm rel}$ to increase in the case of facilitation and to decrease for depression. Between presynaptic action potentials, the release probability decays exponentially back to its "resting" value, P_0,

$$\tau_{\rm P}\frac{dP_{\rm rel}}{dt} = P_0 - P_{\rm rel}\,. \tag{5.37}$$

The parameter $\tau_{\rm P}$ controls the rate at which the release probability decays to P_0.

The models of facilitation and depression differ in how the release probability is changed by presynaptic activity. In the case of facilitation, $P_{\rm rel}$ is augmented by making the replacement $P_{\rm rel} \to P_{\rm rel} + f_{\rm F}(1 - P_{\rm rel})$ immediately after a presynaptic action potential (as in equation 5.32). The parameter $f_{\rm F}$ (with $0 \le f_{\rm F} \le 1$) controls the degree of facilitation, and the factor $(1 - P_{\rm rel})$ prevents the release probability from growing larger than 1. To model depression, the release probability is reduced after a presynaptic action potential by making the replacement $P_{\rm rel} \to f_{\rm D}P_{\rm rel}$. In this case, the parameter $f_{\rm D}$ (with $0 \le f_{\rm D} \le 1$) controls the amount of depression, and the factor $P_{\rm rel}$ prevents the release probability from becoming negative.

We begin by analyzing the effects of facilitation on synaptic transmission for a presynaptic spike train with Poisson statistics. In particular, we compute the average steady-state release probability, denoted by $\langle P_{\rm rel}\rangle$. $\langle P_{\rm rel}\rangle$ is determined by requiring that the facilitation that occurs after each presynaptic action potential is exactly canceled by the average exponential decrement that occurs between presynaptic spikes. Consider two presynaptic action potentials separated by an interval τ, and suppose that the release probability takes its average value $\langle P_{\rm rel}\rangle$ at the time of the first spike. Immediately after this spike, it is augmented to $\langle P_{\rm rel}\rangle + f_{\rm F}(1 - \langle P_{\rm rel}\rangle)$. By the time of the second spike, this will have decayed to $P_0 + (\langle P_{\rm rel}\rangle + f_{\rm F}(1 - \langle P_{\rm rel}\rangle) - P_0)\exp(-\tau/\tau_{\rm P})$, which is obtained by integrating equation 5.37. The average value of the exponential decay factor in this expression is the integral over all positive τ values of $\exp(-\tau/\tau_{\rm P})$ times the probability density for a Poisson spike train with a firing rate r to produce an interspike interval of duration τ, which is $r\exp(-r\tau)$ (see chapter 1). Thus, the average exponential decrement is

$$r\int_0^\infty d\tau\ \exp(-r\tau - \tau/\tau_{\rm P}) = \frac{r\tau_{\rm P}}{1 + r\tau_{\rm P}}\,. \tag{5.38}$$

In order for the release probability to return, on average, to its steady-state value between presynaptic spikes, we must therefore require that

$$\langle P_{\rm rel}\rangle = P_0 + \left(\langle P_{\rm rel}\rangle + f_{\rm F}(1 - \langle P_{\rm rel}\rangle) - P_0\right)\frac{r\tau_{\rm P}}{1 + r\tau_{\rm P}}\,. \tag{5.39}$$

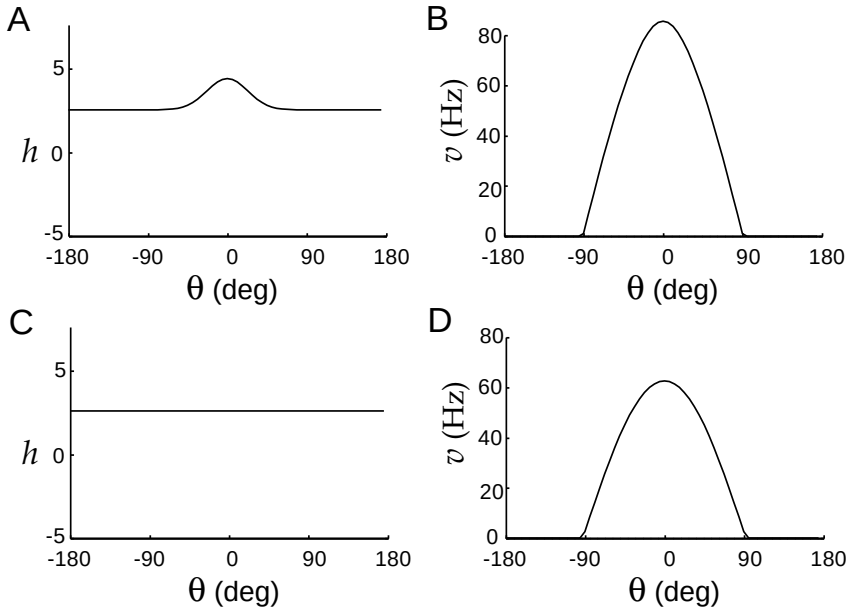

Figure 7.14 Sustained activity in a recurrent network. (A) Input to the neurons of the network consisting of localized excitation and a constant background. (B) The activity of the network neurons in response to the input of panel A. (C) Constant network input. (D) Response to the constant input of panel C when it immediately followed the input in A. The model is the same as that used in figures 7.9, 7.12, and 7.13.

Θ as a memory ($\Theta = 0^\circ$ in the figure). The activities of the units $v(\theta)$ depend on Θ in an essentially nonlinear manner, but if we consider linear perturbations around this nonlinear solution, there is an eigenvector with eigenvalue $\lambda_1 = 1$ associated with shifts in the value of Θ. In this case, it can be shown that $\lambda_1 = 1$ because the network was constructed to be translationally invariant.

Maximum Likelihood and Network Recoding

Recurrent networks can generate characteristic patterns of activity even when they receive complex inputs (figure 7.9), and can maintain these patterns while receiving constant input (figure 7.14). Pouget et al. (1998) suggested that the location of the characteristic pattern (i.e., the value of Θ associated with the peak of the population activity profile) could be interpreted as a match of a fixed template curve to the input activity profile. This curve-fitting operation is at the heart of the maximum likelihood decoding method we described in chapter 3 for estimating a stimulus variable such as Θ. In the maximum likelihood method, the fitting curve is determined by the tuning functions of the neurons, and the curve-fitting procedure is defined by the characteristics of the noise perturbing the input activities. If the properties of the recurrent network match these op-

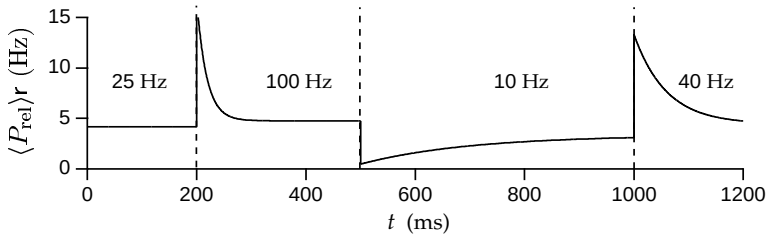

Figure 5.19 The average rate of transmission for a synapse with depression when the presynaptic firing rate changes in a sequence of steps. The firing rates were held constant at the values 25, 100, 10, and 40 Hz, except for abrupt changes at the times indicated by the dashed lines. The parameters of the model are $P_0 = 1$, $f_D = 0.6$, and $\tau_P = 500$ ms.

that depress do not convey information about the values of constant, high presynaptic firing rates to their postsynaptic targets. The presynaptic firing rate at which transmission starts to become independent of r is around $1/((1 - f_D)\tau_P)$.

Figure 5.19 shows the average transmission rate, $\langle P_{\rm rel}\rangle$r, in response to a series of steps in the presynaptic firing rate. Note first that the steady-state transmission rates during the 25, 100, 10, and 40 Hz periods are quite similar. This is a consequence of the $1/r$ dependence of the average release probability, as discussed above. The largest transmission rates in the figure occur during the sharp upward transitions between different presynaptic rates. This illustrates the important point that depressing synapses amplify transient signals relative to steady-state inputs. The transients corresponding the 25 to 100 Hz transition and the 10 to 40 Hz transition are of roughly equal amplitudes, but the transient for the 10 to 40 Hz transition is broader than that for the 25 to 100 Hz transition.

The equality of amplitudes of the two upward transients in figure 5.19 is a consequence of the $1/r$ behavior of $\langle P_{\rm rel}\rangle$. Suppose that the presynaptic firing rate makes a sudden transition from a steady value r to a new value $r + \Delta r$. Before the transition, the average release probability is given by equation 5.42. Immediately after the transition, before the release probability has had time to adjust to the new input rate, the average transmission rate will be this previous value of $\langle P_{\rm rel}\rangle$ times the new rate $r + \Delta r$, which is $P_0(r + \Delta r)/(1 + (1 - f_D)r\tau_P)$. For sufficiently high rates, this is approximately proportional to $(r + \Delta r)/r$. The size of the change in the transmission rate is thus proportional to $\Delta r/r$, which means that depressing synapses not only amplify transient inputs, they transmit them in a scaled manner. The amplitude of the transient transmission rate is proportional to the fractional change, not the absolute change, in the presynaptic firing rate. The two transients seen in figure 5.19 have similar amplitudes because in both cases $\Delta r/r = 3$. The difference in the recovery time for the two upward transients in figure 5.19 is due to the fact that the effective time constant governing the recovery to a new steady-state level r is $\tau_P/(1 + (1 - f_D)\tau_P r)$.

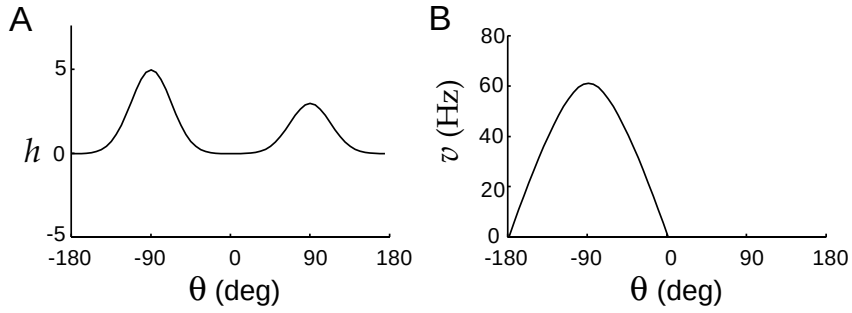

Figure 7.12 Winner-takes-all input selection by a nonlinear recurrent network. (A) The input to the network consisting of two peaks. (B) The output of the network has a single peak at the location of the higher of the two peaks of the input. The model is the same as that used in figure 7.9.

has a single peak at the location of the input bump with the larger amplitude (the one at -90°). This occurs because the nonlinear recurrent network supports the stereotyped unimodal activity pattern seen in figure 7.12B, so a multimodal input tends to generate a unimodal output. The height of the input peak has a large effect in determining where the single peak of the network output is located, but it is not the only feature that determines the response. For example, the network output can favor a broader, lower peak over a narrower, higher one.

Gain Modulation

A nonlinear recurrent network can generate an output that resembles the gain-modulated responses of posterior parietal neurons shown in figure 7.6, as noted by Salinas and Abbott (1996). To obtain this result, we interpret the angle θ as a preferred direction in the visual field in retinal coordinates (the variable we called s earlier in the chapter). The signal corresponding to gaze direction (what we called g before) is represented as a constant input to all neurons irrespective of their preferred stimulus angle. Figure 7.13 shows the effect of adding such a constant term to the input of the nonlinear network. The input shown in figure 7.13A corresponds to a visual target located at a retinal position of 0°. The different lines show different values of the constant input, representing three different gaze directions.

The responses shown in figure 7.13B all have localized activity centered around $\theta = 0^\circ$, indicating that the individual neurons have fixed tuning curves expressed in retinal coordinates. The effect of the constant input, representing gaze direction, is to scale up or gain-modulate these tuning curves, producing a result similar to that shown in figure 7.6. The additive constant in the input shown in figure 7.13A has a multiplicative effect on the output activity shown in 7.13B. This is primarily due to the fact that the width of the activity profiles is fixed by the recurrent network interaction,

the term $r_m \overline{g}_s P_s V$ changes the membrane conductance. The effects of the latter term are referred to as shunting, and they can be identified most easily if we divide equation 5.43 by $1 + r_m \overline{g}_s P_s$ to obtain

$$\frac{\tau_m}{1 + r_m \overline{g}_s P_s} \frac{dV}{dt} = -V + \frac{E_L + r_m \overline{g}_s P_s E_s + R_m I_e}{1 + r_m \overline{g}_s P_s}. \tag{5.44}$$

The shunting effects of the synapse are seen in this equation as a decrease in the effective membrane time constant, and a divisive reduction in the impact of the leakage and synaptic reversal potentials and of the electrode current.

The shunting effects seen in equation 5.44 have been proposed as a possible basis for neural computations involving division. However, shunting has a divisive effect only on the membrane potential of an integrate-and-fire neuron; its effect on the firing rate is subtractive. To see this, assume that synaptic input is arriving at a sufficient rate to maintain a relatively constant value of P_s. In this case, shunting amounts to changing the value of the membrane resistance from R_m to $R_m/(1 + r_m \overline{g}_s P_s)$. Recalling equation 5.12 for the firing rate of the integrate-and-fire model, and the fact that $\tau_m = C_m R_m$, we can write the firing rate in a form that reveals its dependence on R_m,

$$r_{isi} \approx \left[\frac{E_L - V_{th}}{C_m R_m (V_{th} - V_{reset})} + \frac{I_e}{C_m (V_{th} - V_{reset})} \right]_+ . \tag{5.45}$$

Changing R_m modifies only the constant term in this equation; it has no effect on the dependence of the firing rate on I_e.

Regular and Irregular Firing Modes

Integrate-and-fire models are useful for studying how neurons sum large numbers of synaptic inputs and how networks of neurons interact. One issue that has received considerable attention is the degree of variability in the firing output of integrate-and-fire neurons receiving synaptic input. This work has led to the realization that neurons can respond to multiple synaptic inputs in two different modes of operation depending on the balance that exists between excitatory and inhibitory contributions.

The two modes of operation are illustrated in figure 5.21, which shows membrane potentials of an integrate-and-fire model neuron responding to 1000 excitatory and 200 inhibitory inputs. Each input consists of an independent Poisson spike train driving a synaptic conductance. The upper panels of figure 5.21 show the membrane potential with the action potential generation mechanism of the model turned off, and figures 5.21A and 5.21B illustrate the two different modes of operation. In figure 5.21A, the effect of the excitatory inputs is strong enough, relative to that of the inhibitory inputs, to make the average membrane potential, when action

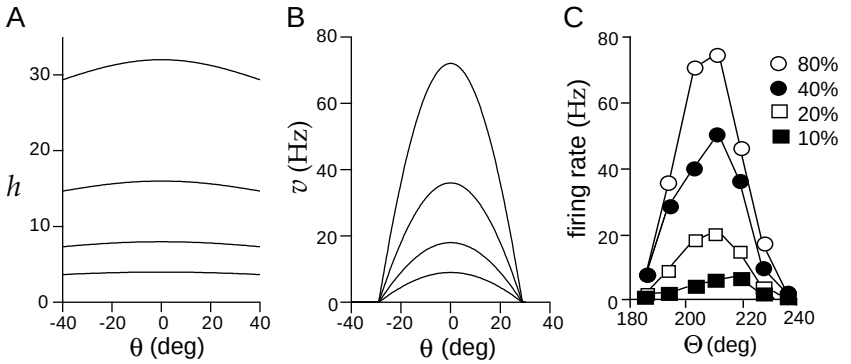

Figure 7.10 The effect of contrast on orientation tuning. (A) The feedforward input as a function of preferred orientation. The four curves, from top to bottom, correspond to contrasts of 80%, 40%, 20%, and 10%. (B) The output firing rates in response to different levels of contrast as a function of orientation preference. These are also the response tuning curves of a single neuron with preferred orientation 0. As in A, the four curves, from top to bottom, correspond to contrasts of 80%, 40%, 20%, and 10%. The recurrent model had $\lambda_0 = 7.3$, $\lambda_1 = 11$, $A = 40$ Hz, and $\epsilon = 0.1$. (C) Tuning curves measured experimentally at four contrast levels, as indicated in the legend. (C adapted from Sompolinsky and Shapley, 1997 based on data from Sclar and Freeman, 1982.)

tuning curve can be reduced by including a positive threshold in the response function of equation 7.34, or by changing the amount of inhibition, but it stays roughly constant as a function of stimulus strength.

A Recurrent Model of Complex Cells in Primary Visual Cortex

In the model of orientation tuning discussed in the previous section, recurrent amplification enhances selectivity. If the pattern of network connectivity amplifies nonselective rather than selective responses, recurrent interactions can also decrease selectivity. Recall from chapter 2 that neurons in the primary visual cortex are classified as simple or complex, depending on their sensitivity to the spatial phase of a grating stimulus. Simple cells respond maximally when the spatial positioning of the light and dark regions of a grating matches the locations of the ON and OFF regions of their receptive fields. Complex cells do not have distinct ON and OFF regions in their receptive fields, and respond to gratings of the appropriate orientation and spatial frequency relatively independently of where their light and dark stripes fall. In other words, complex cells are insensitive to spatial phase.

Chance, Nelson, and Abbott (1999) showed that complex cell responses could be generated from simple cell responses by a recurrent network. As in chapter 2, we label spatial phase preferences by the angle ϕ. The feedforward input $h(\phi)$ in the model is set equal to the rectified response of a simple cell with preferred spatial phase ϕ (figure 7.11A). Each neuron in the network is labeled by the spatial phase preference of its feedfor-

only when there is a fluctuation in the total synaptic input strong enough to make the membrane potential reach the threshold. This produces an irregular spike train, such as that seen in the lower panel of figure 5.21B, which has a C_V value of 0.84.

The high degree of variability seen in the spiking patterns of in vivo recordings of cortical neurons (see chapter 1) suggests that they are better approximated by an integrate-and-fire model operating in an irregular-firing mode. There are advantages to operating in the irregular-firing mode that may compensate for its increased variability. One is that neurons firing in the irregular mode reflect in their outputs the temporal properties of fluctuations in their total synaptic input. In the regular firing mode, the timing of output spikes is only weakly related to the temporal character of the input spike trains. In addition, neurons operating in the irregular firing mode can respond more quickly to changes in presynaptic spiking patterns and firing rates than those operating in the regular firing mode.

5.10 Chapter Summary

In this chapter, we considered the basic electrical properties of neurons, including their intracellular and membrane resistances, capacitances, and active voltage-dependent and synaptic conductances. We introduced the Nernst equation for equilibrium potentials and the formalism of Hodgkin and Huxley for describing persistent, transient, and hyperpolarization-activated conductances. Methods were introduced for modeling stochastic channel opening and stochastic synaptic transmission, including the effects of synaptic facilitation and depression. We discussed a number of ways of describing synaptic conductances following the release of a neurotransmitter. Two models of action potential generation were discussed, the simple integrate-and-fire scheme and the more realistic Hodgkin-Huxley model.

5.11 Appendices

A: Integrating the Membrane Potential

We begin by considering the numerical integration of equation 5.8. It is convenient to rewrite this equation in the form

$$\tau_V \frac{dV}{dt} = V_\infty - V\,, \tag{5.46}$$

where $\tau_V = \tau_m$ and $V_\infty = E_\mathrm{L} + R_\mathrm{m} I_\mathrm{e}$. When the electrode current I_e is independent of time, the solution of this equation is

$$V(t) = V_\infty + (V(t_0) - V_\infty)\exp(-(t-t_0)/\tau_V)\,, \tag{5.47}$$

generates a much smoother output response profile (figure 7.9B). The output response of the rectified network corresponds roughly to the positive part of the sinusoidal response profile of the linear network (figure 7.8B). The negative output has been eliminated by the rectification. Because fewer neurons in the network have nonzero responses than in the linear case, the value of the parameter λ_1 in equation 7.33 has been increased to 1.9. This value, being larger than 1, would lead to an unstable network in the linear case. While nonlinear networks can also be unstable, the restriction to eigenvalues less than 1 is no longer the relevant condition.

In a nonlinear network, the Fourier analysis of the input and output responses is no longer as informative as it is for a linear network. Due to the rectification, the $\nu = 0$, 1, and 2 Fourier components are all amplified (figure 7.9D) compared to their input values (figure 7.9C). Nevertheless, except for rectification, the nonlinear recurrent network amplifies the input signal selectively in a manner similar to the linear network.

A Recurrent Model of Simple Cells in Primary Visual Cortex

In chapter 2, we discussed a feedforward model in which the elongated receptive fields of simple cells in primary visual cortex were formed by summing the inputs from neurons of the lateral geniculate nucleus (LGN) with their receptive fields arranged in alternating rows of ON and OFF cells. While this model quite successfully accounts for a number of features of simple cells, such as orientation tuning, it is difficult to reconcile with the anatomy and circuitry of the cerebral cortex. By far the majority of the synapses onto any cortical neuron arise from other cortical neurons, not from thalamic afferents. Therefore, feedforward models account for the response properties of cortical neurons while ignoring the inputs that are numerically most prominent. The large number of intracortical connections suggests, instead, that recurrent circuitry might play an important role in shaping the responses of neurons in primary visual cortex.

Ben-Yishai, Bar-Or, and Sompolinsky (1995) developed a model in which orientation tuning is generated primarily by recurrent rather than feedforward connections. The model is similar in structure to the model of equations 7.35 and 7.33, except that it includes a global inhibitory interaction. In addition, because orientation angles are defined over the range from $-\pi/2$ to $\pi/2$, rather than over the full 2π range, the cosine functions in the model have extra factors of 2 in them. The basic equation of the model, as we implement it, is

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \int_{-\pi/2}^{\pi/2} \frac{d\theta'}{\pi} \left(-\lambda_0 + \lambda_1 \cos(2(\theta - \theta')) \right) v(\theta') \right]_+ , \tag{7.36}$$

where $v(\theta)$ is the firing rate of a neuron with preferred orientation θ.

constant over this interval if any of the conductances are Ca^{2+}-dependent). Then, τ_z and z_∞, which are functions of V (and possibly $[Ca^{2+}]$) can be treated as constants over this period and z can be updated by a rule identical to 5.48,

$$z(t + \Delta t) = z_\infty + (z(t) - z_\infty) \exp(-\Delta t/\tau_z). \tag{5.52}$$

An efficient integration scheme for conductance-based models is to alternate using rule (5.48) to update the membrane potential and rule (5.52) to update all the gating variables. It is important to alternate the updating of V with that of the gating variables, rather than doing them all simultaneously, as this keeps the method accurate to second order in Δt. If Ca^{2+}-dependent conductances are included, the intracellular Ca^{2+} concentration should be computed simultaneously with the membrane potential. By alternating the updating, we mean that the membrane potential is computed at times $0, \Delta t, 2\Delta t, \ldots$, while the gating variables are computed at times $\Delta t/2, 3\Delta t/2, 5\Delta t/2, \ldots$. A discussion of the second-order accuracy of this scheme is given in Mascagni and Sherman (1998).

5.12 Annotated Bibliography

Jack et al. (1975), **Tuckwell (1988)**, **Johnston & Wu (1995)**, **Koch & Segev (1998)**, and **Koch (1998)** cover much of the material in this chapter and in chapter 6. **Hille (1992)** provides a comprehensive treatment of ion channels. **Hodgkin & Huxley (1952)** presents the classic biophysical model of the action potential, and **Sakmann & Neher (1983)** describes patch clamp recording techniques allowing single channels to be studied electrophysiologically.

The integrate-and-fire model was introduced by Lapicque (1907). **Destexhe et al. (1994)** describes kinetic models of both ion channels and short-term postsynaptic effects at synapses. Marom & Abbott (1994) shows how the Na^+ channel model of Patlak (1991) can be reconciled with typical macroscopic conductance models. For a review of the spike-response model, the integrated version of the integrate-and-fire model, see **Gerstner (1998)**. Wang (1994) analyzes a spike-rate adaptation model similar to the one we presented, and Stevens & Zador (1998) introduces an integrate-and-fire model with time-dependent parameters.

The dynamic aspects of synaptic transmission are reviewed in **Magleby (1987)** and **Zucker (1989)**. Our presentation followed Abbott et al. (1997), Varela et al. (1997), and Tsodyks & Markram (1997). For additional implications of short-term synaptic plasticity for cortical processing, see Lisman (1997) and Chance et al. (1998). Wang & Rinzel (1992) notes that inhibitory synapses can synchronize coupled cells, and in our discussion we followed the treatment in van Vreeswijk et al. (1994). Our analysis of the regular and irregular firing mode regimes of integrate-and-fire cells was

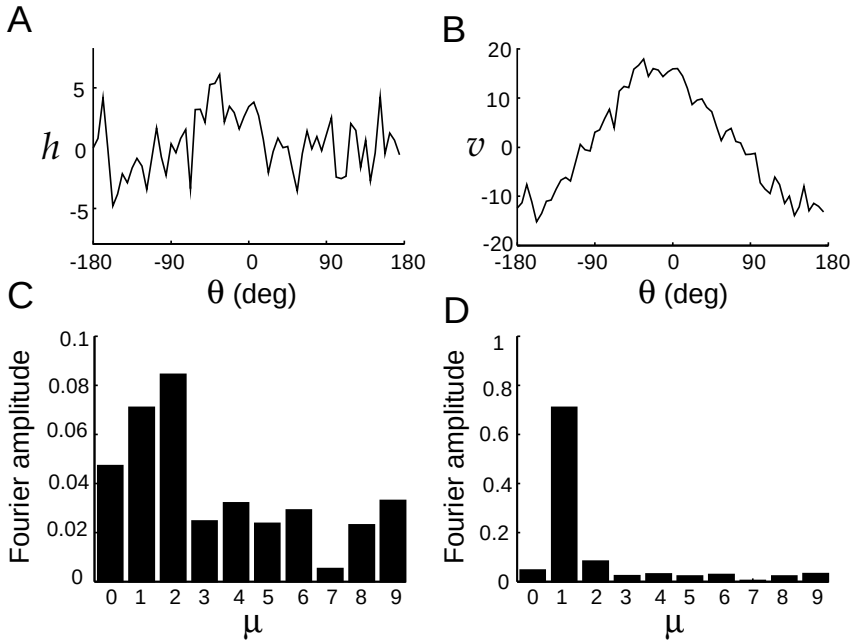

Figure 7.8 Selective amplification in a linear network. (A) The input to the neurons of the network as a function of their preferred stimulus angle. (B) The activity of the network neurons plotted as a function of their preferred stimulus angle in response to the input of panel A. (C) The Fourier transform amplitudes of the input shown in panel A. (D) The Fourier transform amplitudes of the output shown in panel B. The recurrent coupling of this network model took the form of equation 7.33 with $\lambda_1 = 0.9$. (This figure, and figures 7.9, 7.12, 7.13, and 7.14, were generated using software from Carandini and Ringach, 1997.)

be quantified by comparing the Fourier amplitude of v_∞, for a given μ value, with the analogous amplitude for the input h. According to equation 7.32, the ratio of these quantities is $1/(1-\lambda_\mu)$, so, in this case, the $\mu = 1$ amplitude should be amplified by a factor of 10 while all other amplitudes are unamplified. This factor of 10 amplification can be seen by comparing the $\mu = 1$ Fourier amplitudes in figures 7.8C and D (note the different scales for the vertical axes). All the other components are unamplified. As a result, the output of the network is primarily in the form of a cosine function with $\mu = 1$, as seen in figure 7.8B.

Nonlinear Recurrent Networks

A linear model does not provide an adequate description of the firing rates of a biological neural network. The most significant problem is that the firing rates in a linear network can take negative values. This problem can be fixed by introducing rectification into equation 7.11 by choosing

rectification

$$\mathbf{F}(\mathbf{h} + \mathbf{M} \cdot \mathbf{r}) = [\mathbf{h} + \mathbf{M} \cdot \mathbf{r} - \boldsymbol{\gamma}]_+ , \tag{7.34}$$

6 Model Neurons II: Conductances and Morphology

6.1 Levels of Neuron Modeling

In modeling neurons, we must deal with two types of complexity: the intricate interplay of active conductances that makes neuronal dynamics so rich and interesting, and the elaborate morphology that allows neurons to receive and integrate inputs from so many other neurons. The first part of this chapter extends the material presented in chapter 5 by examining single-compartment models with a wider variety of voltage-dependent conductances, and hence a wider range of dynamic behaviors, than the Hodgkin-Huxley model. In the second part of the chapter, we introduce methods used to study the effects of morphology on the electrical characteristics of neurons. An analytic approach known as cable theory is presented first, followed by a discussion of multi-compartment models that permit numerical simulation of complex neuronal structures.

Model neurons range from greatly simplified caricatures to highly detailed descriptions involving thousands of differential equations. Choosing the most appropriate level of modeling for a given research problem requires a careful assessment of the experimental information available and a clear understanding of the research goals. Oversimplified models can, of course, give misleading results, but excessively detailed models can obscure interesting results beneath inessential and unconstrained complexity.

6.2 Conductance-Based Models

The electrical properties of neurons arise from membrane conductances with a wide variety of properties. The basic formalism developed by Hodgkin and Huxley to describe the Na^+ and K^+ conductances responsible for generating action potentials (discussed in chapter 5) is also used to represent most of the additional conductances encountered in neuron modeling. Models that treat these aspects of ionic conductances, known as

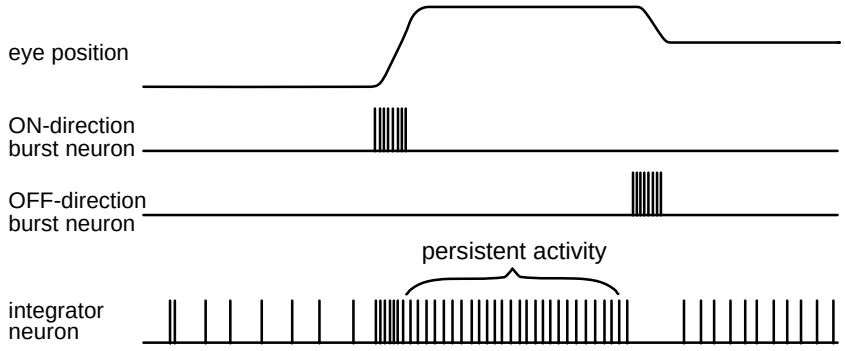

Figure 7.7 Cartoon of burst and integrator neurons involved in horizontal eye positioning. The upper trace represents horizontal eye position during two saccadic eye movements. Motion of the eye is driven by burst neurons that move the eyes in opposite directions (second and third traces from top). The steady-state firing rate (labeled persistent activity) of the integrator neuron is proportional to the time integral of the burst rates, integrated positively for the ON-direction burst neuron and negatively for the OFF-direction burst neuron, and thus provides a memory trace of the maintained eye position. (Adapted from Seung et al., 2000.)

have been discussing have been suggested as models of this system. As outlined in figure 7.7, eye position changes in response to bursts of activity in ocular motor neurons located in the brain stem. Neurons in the medial vestibular nucleus and prepositus hypoglossi appear to integrate these motor signals to provide a persistent memory of eye position. The sustained firing rates of these neurons are approximately proportional to the angular orientation of the eyes in the horizontal direction, and activity persists at an approximately constant rate when the eyes are held fixed (bottom trace in figure 7.7).

The ability of a linear recurrent network to integrate and display persistent activity relies on one of the eigenvalues of the recurrent weight matrix being exactly 1. Any deviation from this value will cause the persistent activity to change over time. Eye position does indeed drift, but matching the performance of the ocular positioning system requires fine-tuning of the eigenvalue to a value extremely close to 1. Including nonlinear interactions does not alleviate the need for a precisely tuned weight matrix. Synaptic modification rules can be used to establish the necessary synaptic weights, but it is not clear how such precise tuning is accomplished in the biological system.

Continuous Linear Recurrent Networks

For a linear recurrent network with continuous labeling, the equation for the firing rate $v(\theta)$ of a neuron with preferred stimulus angle θ is a linear version of equation 7.14,

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + h(\theta) + \rho_\theta \int_{-\pi}^{\pi} d\theta' \, M(\theta - \theta') v(\theta') \, , \qquad (7.29)$$

squid, and we present a multi-compartment simulation of action-potential propagation using this model in a later section. The Connor-Stevens model (Connor and Stevens, 1971; Connor et al. 1977, which is the model we discuss) provides an alternative description of action-potential generation. Like the Hodgkin-Huxley model, it contains fast Na^+, delayed-rectifier K^+, and leakage conductances. The fast Na^+ and delayed-rectifier K^+ conductances have properties somewhat different from those of the Hodgkin-Huxley model, in particular faster kinetics, so the action potentials are briefer. In addition, the Connor-Stevens model contains an extra K^+ conductance, called the A-current, that is transient. K^+ conductances come in wide variety of different forms, and the Connor-Stevens model involves two of them.

A-type potassium current

The membrane current in the Connor-Stevens model is

$$i_{\rm m} = \overline{g}_L(V - E_{\rm L}) + \overline{g}_{\rm Na} m^3 h (V - E_{\rm Na}) + \overline{g}_K n^4 (V - E_{\rm K}) + \overline{g}_{\rm A} a^3 b (V - E_{\rm A}) , \tag{6.4}$$

where $\overline{g}_{\rm L} = 0.003$ mS/mm^2 and $E_{\rm L}$ = -17 mV are the maximal conductance and reversal potential for the leak conductance; and $\overline{g}_{\rm Na} = 1.2$ mS/mm^2, $\overline{g}_{\rm K} = 0.2$ mS/mm^2, $\overline{g}_{\rm A} = 0.477$ mS/mm^2, $E_{\rm Na}$ = 55 mV, $E_{\rm K}$ = -72 mV, and $E_{\rm A}$ = -75 mV (although the A-current is carried by K^+, the model does not require $E_{\rm A} = E_{\rm K}$). The gating variables, m, h, n, a, and b, are determined by equations of the form 6.2 with the gating functions given in appendix A.

The fast Na^+ and delayed-rectifier K^+ conductances generate action potentials in the Connor-Stevens model just as they do in the Hodgkin-Huxley model (see chapter 5). What is the role of the additional A-current? Figure 6.1 illustrates action-potential generation in the Connor-Stevens model. In the absence of an injected electrode current or synaptic input, the membrane potential of the model remains constant at a resting value of -68 mV. For a constant electrode current greater than a threshold value, the model neuron generates action potentials. Figure 6.1A shows how the firing rate of the model depends on the magnitude of the electrode current relative to the threshold value. The firing rate rises continuously from zero and then increases roughly linearly for currents over the range shown. Figure 6.1B shows an example of action-potential generation for one particular value of the electrode current.

Figure 6.1C shows the firing rate as a function of electrode current for the Connor-Stevens model with the maximal conductance of the A-current set to 0. The leakage conductance and reversal potential have been adjusted to keep the resting potential and membrane resistance the same as in the original model. The firing rate is clearly much higher with the A-current turned off. This is because the deinactivation rate of the A-current limits the rise time of the membrane potential between action potentials. In addition, the transition from no firing for currents less than the threshold value to firing with suprathreshold currents is different when the A-current is eliminated. Without the A-current, the firing rate jumps discontinuously to a nonzero value rather than rising continuously. Neurons with firing

The critical feature of this equation is that it involves only one of the coefficients, c_ν. For time-independent inputs $\mathbf{h}$, the solution of equation 7.21 is

$$c_\nu(t) = \frac{\mathbf{e}_\nu \cdot \mathbf{h}}{1-\lambda_\nu}\left(1 - \exp\left(-\frac{t(1-\lambda_\nu)}{\tau_r}\right)\right) + c_\nu(0)\exp\left(-\frac{t(1-\lambda_\nu)}{\tau_r}\right), \tag{7.22}$$

where $c_\nu(0)$ is the value of c_ν at time 0, which is given in terms of the initial firing-rate vector $\mathbf{v}(0)$ by $c_\nu(0) = \mathbf{e}_\nu \cdot \mathbf{v}(0)$.

Equation 7.22 has several important characteristics. If $\lambda_\nu > 1$, the exponential functions grow without bound as time increases, reflecting a fundamental instability of the network. If $\lambda_\nu < 1$, c_ν approaches the steady-state value $\mathbf{e}_\nu \cdot \mathbf{h}/(1-\lambda_\nu)$ exponentially with time constant $\tau_r/(1-\lambda_\nu)$. This steady-state value is proportional to $\mathbf{e}_\nu \cdot \mathbf{h}$, which is the projection of the input vector onto the relevant eigenvector. For $0 < \lambda_\nu < 1$, the steady-state value is amplified relative to this projection by the factor $1/(1-\lambda_\nu)$, which is greater than 1. The approach to equilibrium is slowed relative to the basic time constant τ_r by an identical factor. The steady-state value of $\mathbf{v}(t)$, which we call $\mathbf{v}_\infty$, can be derived from equation 7.19 as

steady state $\mathbf{v}_\infty$

$$\mathbf{v}_\infty = \sum_{\nu=1}^{N_v} \frac{(\mathbf{e}_\nu \cdot \mathbf{h})}{1-\lambda_\nu}\mathbf{e}_\nu\,. \tag{7.23}$$

This steady-state response can also arise from a purely feedforward scheme if the feedforward weight matrix is chosen appropriately, as we invite the reader to verify as an exercise.

Selective Amplification

Suppose that one of the eigenvalues of a recurrent weight matrix, denoted by λ_1, is very close to 1, and all the others are significantly smaller than 1. In this case, the denominator of the $\nu = 1$ term on the right side of equation 7.23 is near 0, and, unless $\mathbf{e}_1 \cdot \mathbf{h}$ is extremely small, this single term will dominate the sum. As a result, we can write

$$\mathbf{v}_\infty \approx \frac{(\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1}{1-\lambda_1}\,. \tag{7.24}$$

Such a network performs selective amplification. The response is dominated by the projection of the input vector along the axis defined by $\mathbf{e}_1$, and the amplitude of the response is amplified by the factor $1/(1-\lambda_1)$, which may be quite large if λ_1 is near 1. The steady-state response of such a network, which is proportional to $\mathbf{e}_1$, therefore encodes an amplified projection of the input vector onto $\mathbf{e}_1$.

Further information can be encoded if more eigenvalues are close to 1. Suppose, for example, that two eigenvectors, $\mathbf{e}_1$ and $\mathbf{e}_2$, have the same

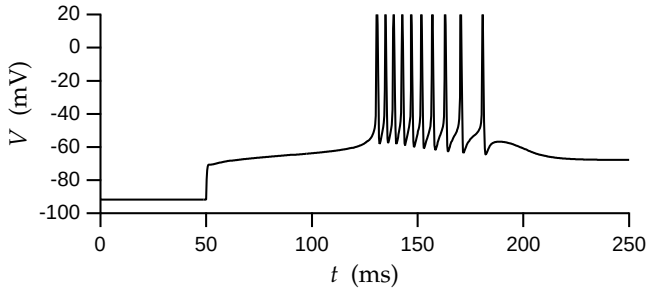

Figure 6.2 A burst of action potentials due to rebound from hyperpolarization. The model neuron was held hyperpolarized for an extended period (until the conductances came to equilibrium) by injection of constant negative electrode current. At $t = 50$ ms, the electrode current was set to 0, and a burst of Na^+ spikes was generated due to an underlying Ca^{2+} spike. The delay in the firing is caused by the presence of the A-current in the model.

data from thalamic relay cells. The membrane current due to the transient Ca^{2+} conductance is expressed as

$$i_{\mathrm{CaT}} = \overline{g}_{\mathrm{CaT}} M^2 H (V - E_{\mathrm{Ca}}) \tag{6.5}$$

with, for the example given here, $\overline{g}_{\mathrm{CaT}} = 0.013$ mS/mm^2 and $E_{\mathrm{Ca}} = 120$ mV. The gating variables for the transient Ca^{2+} conductance are determined from the gating functions in appendix A.

L, T, N and P type Ca^{2+} channels

Several different Ca^{2+} conductances are commonly expressed in neuronal membranes. These are categorized as L, T, N, and P types. L-type Ca^{2+} currents are persistent as far as their voltage dependence is concerned, and they activate at a relatively high threshold. They inactivate due to a Ca^{2+}-dependent rather than voltage-dependent process. T-type Ca^{2+} currents have lower activation thresholds and are transient. N- and P-type Ca^{2+} conductances have intermediate thresholds and are transient and persistent, respectively. They may be responsible for the Ca^{2+} entry that causes the release of transmitter at presynaptic terminals. Entry of Ca^{2+} into a neuron has many secondary consequences ranging from gating Ca^{2+}-dependent channels to inducing long-term modifications of synaptic conductances.

Ca^{2+} spike

burst

A transient Ca^{2+} conductance acts, in many ways, like a slower version of the transient Na^+ conductance that generates action potentials. Instead of producing an action potential, a transient Ca^{2+} conductance generates a slower transient depolarization sometimes called a Ca^{2+} spike. This transient depolarization causes the neuron to fire a burst of action potentials, which are Na^+ spikes riding on the slower Ca^{2+} spike. Figure 6.2 shows such a burst and illustrates one way to produce it. In this example, the model neuron was hyperpolarized for an extended period and then released from hyperpolarization by setting the electrode current to 0. During the prolonged hyperpolarization, the transient Ca^{2+} conductance deinactivated. When the electrode current was set to 0, the resulting depolarization activated the transient Ca^{2+} conductance and generated a burst of

is given by

$$v_\infty = F\left(\rho_\xi \rho_\gamma \int d\xi d\gamma\, w(\xi, \gamma) f_u(s - \xi, g - \gamma)\right). \tag{7.15}$$

For the output neuron to respond to the stimulus location in body-based coordinates, its firing rate must be a function of $s+g$. To see if this is possible, we shift the integration variables in 7.15 by $\xi \to \xi - g$ and $\gamma \to \gamma + g$. Ignoring effects from the end points of the integration (which is valid if s and g are not too close to these limits), we find

$$v_\infty = F\left(\rho_\xi \rho_\gamma \int d\xi d\gamma\, w(\xi - g, \gamma + g) f_u(s + g - \xi, -\gamma)\right). \tag{7.16}$$

This is a function of $s+g$ provided that $w(\xi - g, \gamma + g) = w(\xi, \gamma)$, which holds if $w(\xi, \gamma)$ is a function of the sum $\xi + \gamma$. Thus, the coordinate transformation can be accomplished if the synaptic weight from a given neuron depends only on the sum of its preferred retinal and gaze angles. It has been suggested that weights of this form can arise naturally from random hand and gaze movements through correlation-based synaptic modification of the type discussed in chapter 8.

Figure 7.5C shows responses predicted by equation 7.15 when the synaptic weights are given by a function $w(\xi + \gamma)$. The retinal location of the tuning curve shifts as a function of gaze direction, but would remain stationary if it were plotted instead as a function of $s + g$. This can be seen by noting that the peaks of all three curves in figure 7.5C occur at $s + g = 0$.

Gain-modulated neurons provide a general basis for combining two different input signals in a nonlinear way. In the network we studied, it is possible to find appropriate synaptic weights $w(\xi, \gamma)$ to generate output neuron responses with a wide range of different dependencies on s and g. The mechanism by which sensory and modulatory inputs combine in a multiplicative way in gain-modulated neurons is not known. Later in this chapter, we discuss a recurrent network model for generating gain-modulated responses.

7.4 Recurrent Networks

Recurrent networks have richer dynamics than feedforward networks, but they are more difficult to analyze. To get a feel for recurrent circuitry, we begin by analyzing a linear model, that is, a model for which the relationship between firing rate and synaptic current is linear, $\mathbf{F}(\mathbf{h} + \mathbf{M} \cdot \mathbf{v}) = \mathbf{h} + \mathbf{M} \cdot \mathbf{v}$. The linear approximation is a drastic one that allows, among other things, the components of $\mathbf{v}$ to become negative, which is impossible for real firing rates. Furthermore, some of the features we discuss in connection with linear, as opposed to nonlinear, recurrent networks can also be achieved by a feedforward architecture. Nevertheless, the linear

in the absence of current injection or synaptic input. Periodic bursting is a common feature of neurons in central pattern generators, which are neural circuits that produce periodic patterns of activity to drive rhythmic motor behaviors such as walking, running, or chewing. To illustrate periodic bursting, we consider a model constructed to match the activity of neurons in the crustacean stomatogastric ganglion (STG), a neuronal circuit that controls chewing and digestive rhythms in the foregut of lobsters and crabs. The STG is a model system for investigating the effects of neuromodulators, such as amines and neuropeptides, on the activity patterns of a neural network. Neuromodulators modify neuronal and network behavior by activating, deactivating, or otherwise altering the properties of membrane and synaptic channels. Neuromodulation has a major impact on virtually all neural networks, ranging from peripheral motor pattern generators like the STG to the sensory, motor, and cognitive circuits of the brain.

stomatogastric ganglion

neuromodulator

The model STG neuron contains fast Na^+, delayed-rectifier K^+, A-type K^+, and transient Ca^{2+} conductances similar to those discussed above, although the formulas and parameters used are somewhat different. In addition, the model has a Ca^{2+}-dependent K^+ conductance. Due to the complexity of the model, we do not provide complete descriptions of its conductances except for the Ca^{2+}-dependent K^+ conductance which plays a particularly significant role in the model.

The repolarization of the membrane potential after an action potential is often carried out both by the delayed-rectifier K^+ conductance and by a fast Ca^{2+}-dependent K^+ conductance. Ca^{2+}-dependent K^+ conductances may be voltage dependent, but they are activated primarily by a rise in the level of intracellular Ca^{2+}. A slow Ca^{2+}-dependent K^+ conductance called the after-hyperpolarization (AHP) conductance builds up during sequences of action potentials and typically contributes to the spike-rate adaptation discussed and modeled in chapter 5.

Ca^{2+}-dependent K^+ conductance

after-hyperpolarization conductance

The Ca^{2+}-dependent K^+ current in the model STG neuron is given by

$$i_{\mathrm{KCa}} = \overline{g}_{\mathrm{KCa}} c^4 (V - E_{\mathrm{K}}) , \tag{6.6}$$

where c obeys an equation of the form 6.2, with c_∞ depending on both the membrane potential and the intracellular Ca^{2+} concentration, $[Ca^{2+}]$ (see appendix A). The intracellular Ca^{2+} concentration is computed in this model using a simplified description in which rises in intracellular Ca^{2+} are caused by influx through membrane Ca^{2+} channels, and Ca^{2+} removal is described by an exponential process. The resulting equation for the intracellular Ca^{2+} concentration, $[Ca^{2+}]$, is

$$\frac{d[\mathrm{Ca}^{2+}]}{dt} = -\gamma i_{\mathrm{Ca}} - \frac{[\mathrm{Ca}^{2+}]}{\tau_{\mathrm{Ca}}} . \tag{6.7}$$

Here i_{Ca} is the total Ca^{2+} current per unit area of membrane, τ_{Ca} is the time constant determining the rate at which intracellular Ca^{2+} is removed, and γ is a factor that converts from the electric current due to Ca^{2+} ion flow

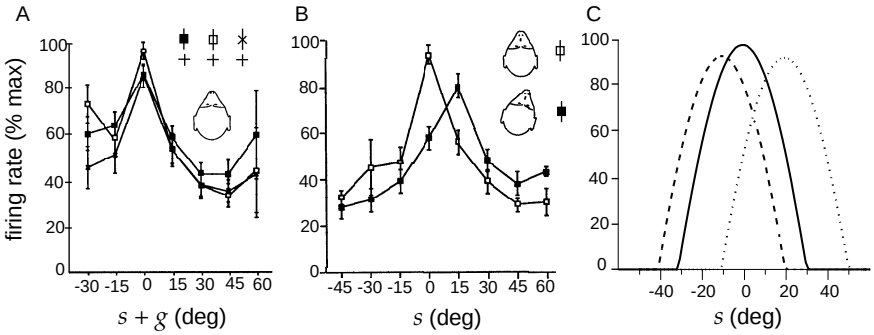

Figure 7.5 Tuning curves of a visually responsive neuron in the premotor cortex of a monkey. Incoming objects approaching at various angles provided the visual stimulation. (A) When the monkey fixated on the three points denoted by the cross symbols, the response tuning curve did not shift with the eyes. In this panel, unlike B and C, the horizontal axis refers to the stimulus location in body-based, not retinal, coordinates ($s + g$, not s). (B) Turning the monkey's head by 15° produced a 15° shift in the response tuning curve as a function of retinal location, indicating that this neuron encoded the stimulus direction in a body-based system. (C) Model tuning curves based on equation 7.15 shift their retinal tuning to remain constant in body-based coordinates. The solid, heavy dashed, and light dashed curves refer to $g = 0°$, $10°$, and $-20°$ respectively. The small changes in amplitude arise from the limited range of preferred retinal location and gaze angles in the model. (A, B adapted from Graziano et al., 1997; C adapted from Salinas and Abbott, 1995.)

are not tied to specific retinal locations but, rather, depend on the relationship of a visual image to various parts of the body. Figures 7.5A and B show tuning curves of a neuron in the premotor cortex of a monkey that responded to visual images of approaching objects. Surprisingly, when the head of the monkey was held stationary during fixation on three different targets, the tuning curves did not shift as the eyes rotated (figure 7.5A). Although the recorded neurons respond to visual stimuli, the responses do not depend directly on the location of the image on the retina. When the head of the monkey is rotated but the fixation point remains the same, the tuning curves shift by precisely the amount of the head rotation (figure 7.5B). Thus, these neurons encode the location of the image in a body-based, not a retinal, coordinate system.

To account for these data, we need to construct a model neuron that is driven by visual input, but that nonetheless has a tuning curve for image location that is not a function of s, the retinal location of the image, but of $s + g$, the location of the object in body-based coordinates. A possible basis for this construction is provided by a combined representation of s and g by neurons in area 7a in the posterior parietal cortex of the monkey. Recordings made in area 7a reveal neurons that fire at rates that depend on both the location of the stimulating image on the retina and the direction of gaze (figure 7.6A). The response tuning curves, expressed as functions of the retinal location of the stimulus, do not shift when the direction of gaze is varied. Instead, shifts of gaze direction affect the magnitude of the visual response. Thus, responses in area 7a exhibit gaze-dependent gain modulation of a retinotopic visual receptive field.

gain modulation

free Ca^{2+} ions in the cell. This factor is a few percent. The minus sign in front of the γ in equation 6.7 is due to the definition of membrane currents as positive in the outward direction.

Figure 6.4 shows the model STG neuron firing action potentials in bursts. As in the models of figures 6.2 and 6.3, the bursts are transient Ca^{2+} spikes with action potentials riding on top of them. The Ca^{2+} current during these bursts causes a dramatic increase in the intracellular Ca^{2+} concentration. This activates the Ca^{2+}-dependent K^+ current, which, along with the inactivation of the Ca^{2+} current, terminates the burst. The interburst interval is determined primarily by the time it takes for the intracellular Ca^{2+} concentration to return to a low value, which deactivates the Ca^{2+}-dependent K^+ current, allowing another burst to be generated. Although figure 6.4 shows that the conductance of the Ca^{2+}-dependent K^+ current reaches a low value immediately after each burst (due to its voltage dependence), this initial dip is too early for another burst to be generated at that point in the cycle.

6.3 The Cable Equation

Single-compartment models describe the membrane potential over an entire neuron with a single variable. Membrane potentials can vary considerably over the surface of the cell membrane, especially for neurons with long and narrow processes, or if we consider rapidly changing membrane potentials. Figure 6.5A shows the delay and attenuation of an action potential as it propagates from the soma out to the dendrites of a cortical pyramidal neuron. Figure 6.5B shows the delay and attenuation of an excitatory postsynaptic potential (EPSP) initiated in the dendrite by synaptic input as it spreads to the soma. Understanding these features is crucial for determining whether and when a given synaptic input will cause a neuron to fire an action potential.

The attenuation and delay within a neuron are most severe when electrical signals travel down the long, narrow, cablelike structures of dendritic or axonal branches. For this reason, the mathematical analysis of signal propagation within neurons is called cable theory. Dendritic and axonal cables are typically narrow enough that variations of the potential in the radial or axial directions are negligible compared to longitudinal variations. Therefore, the membrane potential along a neuronal cable is expressed as a function of a single longitudinal spatial coordinate x and time, $V(x, t)$, and the basic problem is to solve for this potential.

cable theory

Current flows within a neuron due to voltage gradients. In chapter 5, we discussed how the potential difference across a segment of neuronal cable is related to the longitudinal current flowing down the cable. The longitudinal resistance of a cable segment of length Δx and radius a is given by multiplying the intracellular resistivity r_L by Δx and dividing by the cross-sectional area, πa^2, so that $R_L = r_L \Delta x/(\pi a^2)$. The voltage drop

Continuously Labeled Networks

It is often convenient to identify each neuron in a network by using a parameter that describes some aspect of its selectivity rather than the integer label a or b. For example, neurons in primary visual cortex can be characterized by their preferred orientation angles, preferred spatial phases and frequencies, or other stimulus-related parameters (see chapter 2). In many of the examples in this chapter, we consider stimuli characterized by a single angle Θ, which represents, for example, the orientation of a visual stimulus. Individual neurons are identified by their preferred stimulus angles, which are typically the values of Θ for which they fire at maximum rates. Thus, neuron a is identified by an angle θ_a. The weight of the synapse from neuron b or neuron a' to neuron a is then expressed as a function of the preferred stimulus angles θ_b, $\theta_{a'}$ and θ_a of the pre- and postsynaptic neurons, $W_{ab} = W(\theta_a, \theta_b)$ or $M_{aa'} = M(\theta_a, \theta_{a'})$. We often consider cases in which these synaptic weight functions depend only on the difference between the pre- and postsynaptic angles, so that $W_{ab} = W(\theta_a - \theta_b)$ or $M_{aa'} = M(\theta_a - \theta_{a'})$.

density of coverage ρ_θ

In large networks, the preferred stimulus parameters for different neurons will typically take a wide range of values. In the models we consider, the number of neurons is large and the angles θ_a, for different values of a, cover the range from 0 to 2π densely. For simplicity, we assume that this coverage is uniform, so that the density of coverage, the number of neurons with preferred angles falling within a unit range, which we denote by ρ_θ, is constant. For mathematical convenience in these cases, we allow the preferred angles to take continuous values rather than restricting them to the actual discrete values θ_a for $a = 1, 2, \ldots, N$. Thus, we label the neurons by a continuous angle θ and express the firing rate as a function of θ, so that $u(\theta)$ and $v(\theta)$ describe the firing rates of neurons with preferred angles θ. Similarly, the synaptic weight matrices $\mathbf{W}$ and $\mathbf{M}$ are replaced by functions $W(\theta, \theta')$ and $M(\theta, \theta')$ that characterizes the strength of synapses from a presynaptic neuron with preferred angle θ' to a postsynaptic neuron with preferred angle θ in the feedforward and recurrent cases, respectively.

continuous model

If the number of neurons in a network is large and the density of coverage of preferred stimulus values is high, we can approximate the sums in equation 7.10 by integrals over θ'. The number of postsynaptic neurons with preferred angles within a range $\Delta\theta'$ is $\rho_\theta \Delta\theta'$, so, when we take the limit $\Delta\theta' \to 0$, the integral over θ' is multiplied by the density factor ρ_θ. Thus, in the case of continuous labeling of neurons, equation 7.10 becomes (for constant ρ_θ)

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + F\left(\rho_\theta \int_{-\pi}^{\pi} d\theta' \, W(\theta, \theta') u(\theta') + M(\theta, \theta') v(\theta')\right). \tag{7.14}$$

As we did in equation 7.11, we can write the first term inside the integral of this expression as an input function $h(\theta)$. We make frequent use of continuous labeling for network models, and we often approximate sums over neurons by integrals over their preferred stimulus parameters.

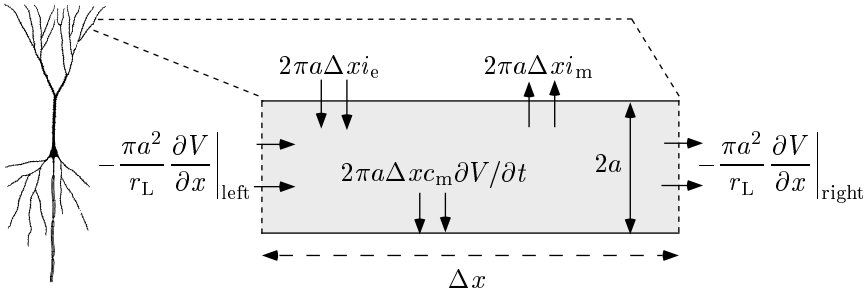

Figure 6.6 The segment of neuron used in the derivation of the cable equation. The longitudinal, membrane, and electrode currents that determine the rate of change of the membrane potential within this segment are denoted. The segment has length Δx and radius a. The expression involving the specific membrane capacitance refers to the rate at which charge builds up on the cell membrane, generating changes in the membrane potential. (The neuron diagram here and in figures 6.15 and 6.16 is from Haberly, 1990.)

All of the currents that can change the membrane potential of the segment being considered are shown in figure 6.6. Current can flow longitudinally into the segment from neighboring segments, and expression 6.8 has been used in figure 6.6 to specify the longitudinal currents at both ends of the segment. Current can flow across the membrane of the segment we are considering through ion and synaptic receptor channels, or through an electrode. The contribution from ion and synaptic channels is expressed as a current per unit area of membrane i_m times the surface area of the segment, $2\pi a\Delta x$. The electrode current is not normally expressed as a current per unit area, but for the present purposes it is convenient to define i_e to be the total electrode current flowing into a given region of the neuronal cable divided by the surface area of that region. The total amount of electrode current being injected into the cable segment of figure 6.6 is then $i_e 2\pi a\Delta x$. Because the electrode current is normally specified by I_e, not by a current per unit area, all the results we obtain will ultimately be re-expressed in terms of I_e. Following the standard convention, membrane and synaptic currents are defined as positive when they are outward, and electrode currents are defined as positive when they are inward.

The cable equation is derived by setting the sum of all the currents shown in figure 6.6 equal to the current needed to charge the membrane. The total longitudinal current entering the cylinder is the difference between the current flowing in on the left and that flowing out on the right. Thus,

$$2\pi a\Delta x c_m \frac{\partial V}{\partial t} = -\left(\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}\right)\Big|_{left} + \left(\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}\right)\Big|_{right} - 2\pi a\Delta x(i_m - i_e)\,. \tag{6.9}$$

Dividing both sides of this equation by $2\pi a\Delta x$, we note that the right side involves the term

$$\frac{1}{\Delta x}\left[\left(\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}\right)\Big|_{right} - \left(\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}\right)\Big|_{left}\right] \to \frac{\partial}{\partial x}\left(\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}\right)\,. \tag{6.10}$$

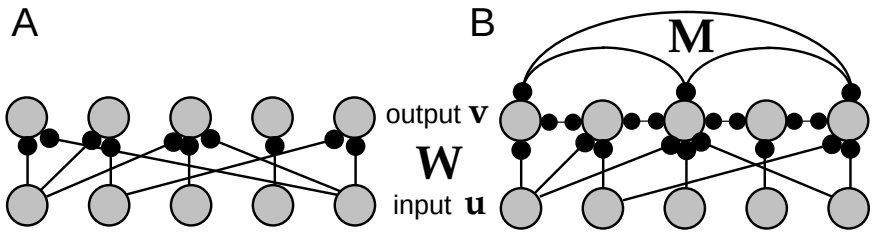

Figure 7.3 Feedforward and recurrent networks. (A) A feedforward network with input rates **u**, output rates **v**, and a feedforward synaptic weight matrix **W**. (B) A recurrent network with input rates **u**, output rates **v**, a feedforward synaptic weight matrix **W**, and a recurrent synaptic weight matrix **M**. Although we have drawn the connections between the output neurons as bidirectional, this does not necessarily imply connections of equal strength in both directions.

vious section with the firing rate given neither by $F(I)$ nor by v but by another function, $G(I, v)$. This compound model provides quite an accurate description of the firing rate of the integrate-and-fire model, but it is more complex than the models used in this chapter.

Feedforward and Recurrent Networks

Figure 7.3 shows examples of network models with feedforward and recurrent connectivity. The feedforward network of figure 7.3A has N_v output units with rates v_a ($a = 1, 2, \ldots, N_v$), denoted collectively by the vector **v**, driven by N_u input units with rates **u**. Equations 7.8 and 7.6 can easily be extended to cover this case by replacing the vector of synaptic weights **w** with a matrix **W**, with the matrix component W_{ab} representing the strength of the synapse from input unit b to output unit a. Using the formulation of equation 7.8, the output firing rates are then determined by

feedforward model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W} \cdot \mathbf{u}) \quad \text{or} \quad \tau_r \frac{dv_a}{dt} = -v_a + F\left(\sum_{b=1}^{N_u} W_{ab} u_b\right). \tag{7.9}$$

We use the notation $\mathbf{W} \cdot \mathbf{u}$ to denote the vector with components $\sum_b W_{ab} u_b$. The use of the dot to represent a sum of a product of two quantities over a shared index is borrowed from the notation for the dot product of two vectors. The expression $\mathbf{F}(\mathbf{W} \cdot \mathbf{u})$ represents the vector with components $F(\sum W_{ab} u_b)$ for $a = 1, 2, \ldots, N_v$.

The recurrent network of figure 7.3B also has two layers of neurons with rates **u** and **v**, but in this case the neurons of the output layer are interconnected with synaptic weights described by a matrix **M**. Matrix element $M_{aa'}$ describes the strength of the synapse from output unit a' to output unit a. The output rates in this case are determined by

recurrent model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v}) \,. \tag{7.10}$$

membrane potential. Eliminating synaptic currents requires us to examine how a neuron responds to the electrode current i_e. In some cases, electrode current can mimic the effects of a synaptic conductance, although the two are not equivalent. In any case, studying responses to electrode current allows us to investigate the effects of different morphologies on membrane potentials.

Typically, a linear approximation for the membrane current is valid only if the membrane potential stays within a limited range, for example, close to the resting potential of the cell. The resting potential is defined as the potential where no net current flows across the membrane. Near this potential, we approximate the membrane current per unit area as

$$i_m = (V - V_{rest})/r_m \,, \tag{6.12}$$

where V_{rest} is the resting potential and r_m is the specific membrane resistance. It is convenient to define v as the membrane potential relative to the resting potential, $v = V - V_{rest}$, so that $i_m = v/r_m$.

$v = V - V_{rest}$

If the radii of the cable segments used to model a neuron are constant except at branches and abrupt junctions, the factor a^2 in equation 6.11 can be taken out of the derivative and combined with the prefactor $1/2ar_L$ to produce a factor $a/2r_L$ that multiplies the spatial second derivative. With this modification and use of the linear expression for the membrane current, the cable equation for v is

$$c_m \frac{\partial v}{\partial t} = \frac{a}{2r_L}\frac{\partial^2 v}{\partial x^2} - \frac{v}{r_m} + i_e \,. \tag{6.13}$$

It is convenient to multiply this equation by r_m, turning the factor that multiplies the time derivative on the left side into the membrane time constant $\tau_m = r_m c_m$. This also changes the expression multiplying the spatial second derivative on the right side of equation 6.13 to $ar_m/2r_L$. This factor has the dimensions of length squared, and it defines a fundamental length constant for a segment of cable of radius a, the electrotonic length,

electrotonic length λ

$$\lambda = \sqrt{\frac{ar_m}{2r_L}} \,. \tag{6.14}$$

Using the values $r_m = 1$ M$\Omega\cdot$mm^2 and $r_L = 1$ k$\Omega\cdot$mm, a cable of radius $a = 2$ μm has an electrotonic length of 1 mm. A segment of cable with radius a and length λ has a membrane resistance that is equal to its longitudinal resistance, as can be seen from equation 6.14,

R_λ

$$R_\lambda = \frac{r_m}{2\pi a \lambda} = \frac{r_L \lambda}{\pi a^2} \,. \tag{6.15}$$

The resistance R_λ defined by this equation is a useful quantity that enters into a number of calculations.

linear cable equation

Expressed in terms of τ_m and λ, the cable equation becomes

$$\tau_m \frac{\partial v}{\partial t} = \lambda^2 \frac{\partial^2 v}{\partial x^2} - v + r_m i_e \,. \tag{6.16}$$

model that is defined by equation 7.6. If instead, $\tau_r \gg \tau_s$, we can make the approximation that equation 7.4 comes to equilibrium quickly compared with equation 7.7. Then we can make the replacement $I_s = \mathbf{w} \cdot \mathbf{u}$ in equation 7.7 and write

firing-rate equation

$$\tau_r \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u}) . \tag{7.8}$$

For most of this chapter, we analyze network models described by the firing-rate dynamics of equation 7.8, although occasionally we consider networks based on equation 7.6.

Firing-Rate Dynamics

The firing-rate models described by equations 7.6 and 7.8 differ in their assumptions about how firing rates respond to and track changes in the input current to a neuron. In one case (equation 7.6), it is assumed that firing rates follow time-varying input currents instantaneously, without attenuation or delay. In the other case (equation 7.8), the firing rate is a low-pass filtered version of the input current. To study the relationship between input current and firing rate, it is useful to examine the firing rate of a spiking model neuron in response to a time-varying injected current, $I(t)$. The model used for this purpose in figure 7.2 is an integrate-and-fire neuron receiving balanced excitatory and inhibitory synaptic input along with a current injected into the soma that is the sum of constant and oscillating components. This model was discussed in chapter 5. The balanced synaptic input is used to represent background input not included in the computation of I_s, and it acts as a source of noise. The noise prevents effects, such as locking of the spiking to the oscillations of the injected current, that would invalidate a firing-rate description.

Figure 7.2 shows the firing rates of the model integrate-and-fire neuron in response to an input current $I(t) = I_0 + I_1 \cos(\omega t)$. The firing rate is plotted at different times during the cycle of the input current oscillations for ω corresponding to frequencies of 1, 50, and 100 Hz. For the panels on the left side, the constant component of the injected current (I_0) was adjusted so the neuron never stopped firing during the cycle. In this case, the relation $v(t) = F(I(t))$ (solid curves) provides an accurate description of the firing rate for all of the oscillation frequencies shown. As long as the neuron keeps firing fairly rapidly, the low-pass filtering properties of the membrane potential are not relevant for the dynamics of the firing rate. Low-pass filtering is irrelevant in this case, because the neuron is continually being shuttled between the threshold and reset values, so it never has a chance to settle exponentially anywhere near its steady-state value.

The right panels in figure 7.2 show that the situation is different if the input current is below the threshold for firing through a significant part

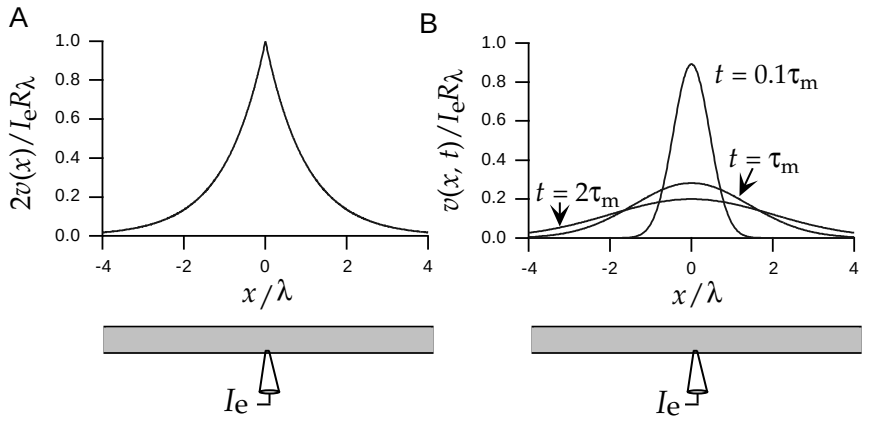

Figure 6.7 The potential for current injection at the point $x = 0$ along an infinite cable. (A) Static solution for a constant electrode current. The potential decays exponentially away from the site of current injection. (B) Time-dependent solution for a δ function pulse of current. The potential is described by a Gaussian function centered at the site of current injection that broadens and shrinks in amplitude over time.

is to determine B, which we do by balancing the current injected with the current that diffuses away from $x = 0$.

In the small region of size Δx around $x = 0$ where the current is injected, the full equation $\lambda^2 d^2v/dx^2 = v - r_m i_e$ must be solved. If the total amount of current injected by the electrode is I_e, the current per unit area injected into this region is $I_e/(2\pi a \Delta x)$. This grows without bound as $\Delta x \to 0$. The first derivative of the membrane potential $v(x) = B\exp(-|x|/\lambda)$ is discontinuous at the point $x = 0$. For small Δx, the derivative at one side of the region we are discussing (at $x = -\Delta x/2$) is approximately B/λ, while at the other side (at $x = +\Delta x/2$) it is $-B/\lambda$. In these expressions, we have used the fact that Δx is small to set $\exp(-|\Delta x|/2\lambda) \approx 1$. For small Δx, the second derivative is approximately the difference between these two first derivatives divided by Δx, which is $-2B/(\lambda \Delta x)$. We can ignore the term v in the cable equation within this small region, because it is not proportional to $1/\Delta x$. Substituting the expressions we have derived for the remaining terms in the equation, we find that $-2\lambda^2 B/(\lambda \Delta x) = -r_m I_e/(2\pi a \Delta x)$, which means that $B = I_e R_\lambda/2$, using R_λ from equation 6.15. Thus, the membrane potential for static current injection at the point $x = 0$ along an infinite cable is

$$v(x) = \frac{I_e R_\lambda}{2} \exp\left(-\frac{|x|}{\lambda}\right). \tag{6.18}$$

According to this result, the membrane potential away from the site of current injection ($x = 0$) decays exponentially with length constant λ (see figure 6.7A). The ratio of the membrane potential at the injection site to the magnitude of the injected current is called the input resistance of the cable. The value of the potential at $x = 0$ is $I_e R_\lambda/2$, indicating that the

As discussed previously, the critical step in the construction of a firing-rate model is the replacement of the neural response function $\rho_b(\tau)$ in equation 7.2 with the firing rate of neuron b, $u_b(\tau)$, so that we write

$$I_s = \sum_{b=1}^{N_u} w_b \int_{-\infty}^{t} d\tau \, K_s(t-\tau) u_b(\tau) \, . \tag{7.3}$$

The synaptic kernel most frequently used in firing-rate models is an exponential, $K_s(t) = \exp(-t/\tau_s)/\tau_s$. With this kernel, we can describe I_s by a differential equation if we take the derivative of equation 7.3 with respect to t,

$$\tau_s \frac{dI_s}{dt} = -I_s + \sum_{b=1}^{N_u} w_b u_b = -I_s + \mathbf{w} \cdot \mathbf{u} \, . \tag{7.4}$$

dot product

In the second equality, we have expressed the sum $\sum w_b u_b$ as the dot product of the weight and input vectors, $\mathbf{w} \cdot \mathbf{u}$. In this and the following chapters, we primarily use the vector versions of equations such as 7.4, but when we first introduce an important new equation, we often write it in its subscripted form as well.

Recall that K describes the temporal evolution of the synaptic current due to both synaptic conductance and dendritic cable effects. For an electrotonically compact dendritic structure, τ_s will be close to the time constant that describes the decay of the synaptic conductance. For fast synaptic conductances such as those due to AMPA glutamate receptors, this may be as short as a few milliseconds. For a long, passive dendritic cable, τ_s may be larger than this, but its measured value is typically quite small.

The Firing Rate

Equation 7.4 determines the synaptic current entering the soma of a postsynaptic neuron in terms of the firing rates of the presynaptic neurons. To finish formulating a firing-rate model, we must determine the postsynaptic firing rate from our knowledge of I_s. For constant synaptic current, the firing rate of the postsynaptic neuron can be expressed as $v = F(I_s)$, where F is the steady-state firing rate as a function of somatic input current. F is also called an activation function. F is sometimes taken to be a saturating function such as a sigmoid function. This is useful in cases where the derivative of F is needed in the analysis of network dynamics. It is also bounded from above, which can be important in stabilizing a network against excessively high firing rates. More often, we use a threshold linear function $F(I_s) = [I_s - \gamma]_+$, where γ is the threshold and the notation $[\,]_+$ denotes half-wave rectification, as in previous chapters. For convenience, we treat I_s in this expression as if it were measured in units of a firing rate (Hz), that is, as if I_s is multiplied by a constant that converts its units from nA to Hz. This makes the synaptic weights dimensionless. The threshold γ also has units of Hz.

activation function $F(I_s)$

threshold γ

we can define a type of "velocity" for this solution by computing the time $t_{\max}$ when the maximum of the potential occurs at a given spatial location. This is done by setting the time derivative of $v(x,t)$ in equation 6.19 to 0, giving

$$t_{\max} = \frac{\tau_m}{4}\left(\sqrt{1+4(x/\lambda)^2}-1\right). \tag{6.20}$$

For large x, $t_{\max} \approx x\tau_m/2\lambda$, corresponding to a velocity of $2\lambda/\tau_m$. For smaller x values, the location of the maximum moves faster than this "velocity" would imply (figure 6.8B).

An Isolated Branching Node

To illustrate the effects of branching on the membrane potential in response to a point source of current injection, we consider a single isolated junction of three semi-infinite cables, as shown in the bottom panels of figure 6.9. For simplicity, we discuss the solution for static current injection at a point, but the results generalize directly to the case of time-dependent currents. We label the potentials along the three segments v_1, v_2, and v_3, and label the distance outward from the junction point along any given segment by the coordinate x (although in figure 6.9 a slightly different convention is used). The electrode injection site is located a distance y away from the junction along segment 2. The solution for the three segments is then

$$\begin{aligned} v_1(x) &= p_1 I_e R_{\lambda_1} \exp(-x/\lambda_1 - y/\lambda_2) \\ v_2(x) &= \frac{I_e R_{\lambda_2}}{2}\left[\exp(-|y-x|/\lambda_2) + (2p_2-1)\exp(-(y+x)/\lambda_2)\right] \\ v_3(x) &= p_3 I_e R_{\lambda_3} \exp(-x/\lambda_3 - y/\lambda_2), \end{aligned} \tag{6.21}$$

where, for $i = 1$, 2, and 3,

$$p_i = \frac{a_i^{3/2}}{a_1^{3/2}+a_2^{3/2}+a_3^{3/2}} \quad , \quad \lambda_i = \sqrt{\frac{a_i r_m}{2r_L}} \quad \text{, and} \quad R_{\lambda_i} = \frac{r_L \lambda_i}{\pi a_i^2}. \tag{6.22}$$

Note that the distances x and y appearing in the exponential functions are divided by the electrotonic length of the segment along which the potential is measured or the current is injected. This solution satisfies the cable equation, because it is constructed by combining solutions of the form 6.18. The only term that has a discontinuous first derivative within the range being considered is the first term in the expression for v_2, and this solves the cable equation at the current injection site because it is identical to 6.18. We leave it to the reader to verify that this solution satisfies the boundary conditions $v_1(0) = v_2(0) = v_3(0)$ and $\sum a_i^2 \partial v_i/\partial x = 0$.

Figure 6.9 shows the potential near a junction where a cable of radius 2 μ breaks into two thinner cables of radius 1 μ. In figure 6.9A, current is injected along the thicker cable, and in figure 6.9B it is injected along one of the thinner branches. In both cases, the site of current injection is one

tered by the dynamics of the conductance changes that each presynaptic action potential evokes in the postsynaptic neuron (see chapter 5) and the dynamics of propagation of the current from the synapse to the soma. The temporal averaging provided by slow synaptic or membrane dynamics can reduce the effects of spike-train variability and help justify the approximation of using firing rates instead of presynaptic spike trains. Firing-rate models are more accurate if the network being modeled has a significant amount of synaptic transmission that is slow relative to typical presynaptic interspike intervals.

synaptic current I_s

The construction of a firing-rate model proceeds in two steps. First, we determine how the total synaptic input to a neuron depends on the firing rates of its presynaptic afferents. This is where we use firing rates to approximate neural response functions. Second, we model how the firing rate of the postsynaptic neuron depends on its total synaptic input. Firing-rate response curves are typically measured by injecting current into the soma of a neuron. We therefore find it most convenient to define the total synaptic input as the total current delivered to the soma as a result of all the synaptic conductance changes resulting from presynaptic action potentials. We denote this total synaptic current by I_s. We then determine the postsynaptic firing rate from I_s. In general, I_s depends on the spatially inhomogeneous membrane potential of the neuron, but we assume that, other than during action potentials or transient hyperpolarizations, the membrane potential remains close to, but slightly below, the threshold for action potential generation. An example of this type of behavior is seen in the upper panels of figure 7.2. I_s is then approximately equal to the synaptic current that would be measured from the soma in a voltage-clamp experiment, except for a reversal of sign. In the next section, we model how I_s depends on presynaptic firing rates.

input rate u
output rate v

In the network models we consider, both the output from, and the input to, a neuron are characterized by firing rates. To avoid a proliferation of sub- and superscripts on the quantity $r(t)$, we use the letter u to denote a presynaptic firing rate, and v to denote a postsynaptic rate. Note that v is used here to denote a firing rate, not a membrane potential. In addition, we use these two letters to distinguish input and output firing rates in network models, a convention we retain through the remaining chapters.

input rate vector $\mathbf{u}$
output rate vector $\mathbf{v}$

When we consider multiple input or output neurons, we use vectors $\mathbf{u}$ and $\mathbf{v}$ to represent their firing rates collectively, with the components of these vectors representing the firing rates of the individual input and output units.

The Total Synaptic Current

Consider a neuron receiving N_u synaptic inputs labeled by $b = 1, 2, \ldots, N_u$ (figure 7.1). The firing rate of input b is denoted by u_b, and the input rates are represented collectively by the N_u-component vector $\mathbf{u}$. We model how the synaptic current I_s depends on presynaptic firing rates by first consid-

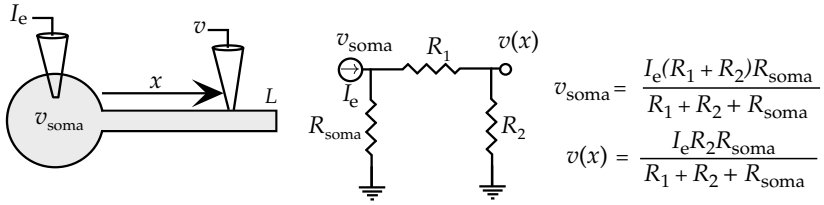

Figure 6.10 The Rall model with static current injected into the soma. The schematic at left shows the recording setup. The potential is measured at the soma and at a distance x along the equivalent cable. The central diagram is the equivalent circuit for this case, and the corresponding formulas for the somatic and dendritic voltages are given at the right. The symbols at the bottom of the resistances R_{soma} and R_2 indicate that $v = 0$ at these points. R_{soma} is the membrane resistance of the soma, and R_1 and R_2 are the resistances given in equations 6.23 and 6.24.

effect of these inputs is usually measured from the soma, and the spike-initiation region of the axon that determines whether the neuron fires an action potential is typically located near the soma. In Rall's model, a compact soma region (represented by one compartment) is connected to a single equivalent cylindrical cable that replaces the entire dendritic region of the neuron (see the schematics in figures 6.10 and 6.12). The critical feature of the model is the choice of the radius and length for the equivalent cable to best match the properties of the dendritic structure being approximated.

The radius a and length L of the equivalent cable are determined by matching two important elements of the full dendritic tree. These are its average length in electrotonic units, which determines the amount of attenuation, and the total surface area, which determines the total membrane resistance and capacitance. The average electrotonic length of a dendrite is determined by considering direct paths from the soma to the terminals of the dendrite. The electrotonic lengths for these paths are constructed by measuring the distance traveled along each of the cable segments traversed in units of the electrotonic length constant for that segment. In general, the total electrotonic length measured by summing these electrotonic segment lengths depends on which terminal of the tree is used as the end point. However, an average value can be used to define an electrotonic length for the full dendritic structure. The length L of the equivalent cable is then chosen so that L/λ is equal to this average electrotonic length, where λ is the length constant for the equivalent cable. The radius of the equivalent cable, which is needed to compute λ, is determined by setting the surface area of the equivalent cable, $2\pi aL$, equal to the surface area of the full dendritic tree.

Under some restrictive circumstances the equivalent cable reproduces the effects of a full tree exactly. Among these conditions is the requirement $a_1^{3/2} = a_2^{3/2} + a_3^{3/2}$ on the radii of any three segments being joined at a node within the tree. Note from equation 6.22 that this condition makes $p_1 = p_2 + p_3 = 1/2$. However, even when the so-called 3/2 law is not exact, the equivalent cable is an extremely useful and often reasonably accurate simplification.

fibers between connected regions are typically comparable, and recurrent synapses typically outnumber feedforward or top-down inputs. We begin this chapter by studying networks with purely feedforward input and then study the effects of recurrent connections. The analysis of top-down connections, for which it is more difficult to establish clear computational roles, is left until chapter 10.

The most direct way to simulate neural networks is to use the methods discussed in chapters 5 and 6 to synaptically connect model spiking neurons. This is a worthwhile and instructive enterprise, but it presents significant computational, calculational, and interpretational challenges. In this chapter, we follow a simpler approach and construct networks of neuron-like units with outputs consisting of firing rates rather than action potentials. Spiking models involve dynamics over time scales ranging from channel openings that can take less than a millisecond, to collective network processes that may be several orders of magnitude slower. Firing-rate models avoid the short time scale dynamics required to simulate action potentials and thus are much easier to simulate on computers. Firing-rate models also allow us to present analytic calculations of some aspects of network dynamics that could not be treated in the case of spiking neurons. Finally, spiking models tend to have more free parameters than firing-rate models, and setting these appropriately can be difficult.

There are two additional arguments in favor of firing-rate models. The first concerns the apparent stochasticity of spiking. The models discussed in chapters 5 and 6 produce spike sequences deterministically in response to injected current or synaptic input. Deterministic models can predict spike sequences accurately only if all their inputs are known. This is unlikely to be the case for the neurons in a complex network, and network models typically include only a subset of the many different inputs to individual neurons. Therefore, the greater apparent precision of spiking models may not actually be realized in practice. If necessary, firing-rate models can be used to generate stochastic spike sequences from a deterministically computed rate, using the methods discussed in chapters 1 and 2.

The second argument involves a complication with spiking models that arises when they are used to construct simplified networks. Although cortical neurons receive many inputs, the probability of finding a synaptic connection between a randomly chosen pair of neurons is actually quite low. Capturing this feature, while retaining a high degree of connectivity through polysynaptic pathways, requires including a large number of neurons in a network model. A standard way of dealing with this problem is to use a single model unit to represent the average response of several neurons that have similar selectivities. These "averaging" units can then be interconnected more densely than the individual neurons of the actual network, so fewer of them are needed to build the model. If neural responses are characterized by firing rates, the output of the model unit is simply the average of the firing rates of the neurons it represents collectively. However, if the response is a spike, it is not clear how the spikes of the represented neurons can be averaged. The way spiking models are

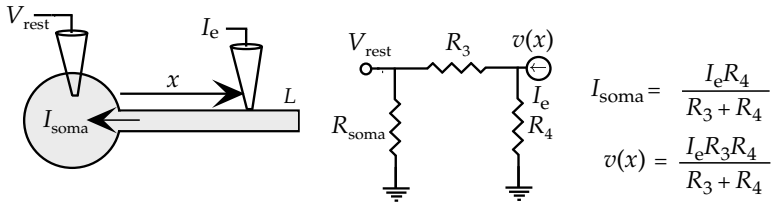

Figure 6.12 The Rall model with static current injected a distance x along the equivalent cable while the soma is clamped at its resting potential. The schematic at left shows the recording setup. The potential at the site of the current injection and the current entering the soma are measured. The central diagram is the equivalent circuit for this case, and the corresponding formulas for the somatic current and dendritic voltage are given at the right. R_{soma} is the membrane resistance of the soma, and R_3 and R_4 are the resistances given in equations 6.26 and 6.27.

This result is plotted in figure 6.11.

Figure 6.12 shows the equivalent circuit for the Rall model when current is injected at a location x along the cable, and the soma is clamped at $v_{soma} = 0$ (or equivalently $V_{soma} = V_{rest}$). The equivalent circuit can be used to determine the current entering the soma and the voltage at the site of current injection. In this case, the somatic resistance is irrelevant because the soma is clamped at its resting potential. The other resistances are

$$R_3 = R_\lambda \sinh(x/\lambda) \tag{6.26}$$

and

$$R_4 = \frac{R_\lambda \sinh(x/\lambda)\cosh((L-x)/\lambda)}{\cosh(L/\lambda) - \cosh((L-x)/\lambda)}. \tag{6.27}$$

The input resistance for this configuration, as measured from the dendrite, is determined by R_3 and R_4 acting in parallel, and is $R_3 R_4/(R_3 + R_4) = R_\lambda \sinh(x/\lambda)\cosh((L-x)/\lambda)/\cosh(L/\lambda)$. When L and x are both much larger than λ, this approaches the limiting value R_λ. The current attenuation is defined as the ratio of the somatic current to the electrode current, and is given by

$$\frac{I_{soma}}{I_e} = \frac{R_4}{R_3 + R_4} = \frac{\cosh((L-x)/\lambda)}{\cosh(L/\lambda)}. \tag{6.28}$$

The inward current attenuation (plotted in figure 6.11) for the recording configuration of figure 6.12 is identical to the outward voltage attenuation for figure 6.10 given by equation 6.25. Equality of the voltage attenuation measured in one direction and the current attenuation measured in the opposite direction is a general feature of linear cable theory.

The Morphoelectrotonic Transform

The membrane potential for a neuron of complex morphology is obviously much more difficult to compute than the simple cases we have

cable to define all the c' and f' parameters. Then solve for ΔV_N from equation 6.56 and iterate back up the cable, solving for the ΔV's using 6.53. This process takes only $2N$ steps.

We leave the extension of this method to the case of a branched cable as an exercise for the reader. The general procedure is similar to the one we presented for a nonbranching cable. The equations are solved by starting at the ends of the branches and moving in toward their branching node, then continuing on as for a nonbranching cable, and finally reversing direction and completing the solution moving in the opposite direction along the cable and its branches.

6.7 Annotated Bibliography

Many of the references for chapter 5 apply to this chapter as well, including **Jack et al. (1975)**, **Tuckwell (1988)**, **Johnston & Wu (1995)**, **Koch & Segev (1998)**, **Koch (1998)**, **Hille (1992)**, and **Mascagni & Sherman (1998)**. **Rall (1977)** describes cable theory, the equivalent cable model of dendritic trees, and the 3/2 law. The solution of equation 6.21 can be constructed using the set of rules for solving the linear cable equation on arbitrary trees found in Abbott (1992; see also Abbott et al., 1991). **Marder & Calabrese (1996)** reviews neuromodulation.

Two freely available software packages for detailed neuronal modeling are in wide use, `Neuron` (see Hines & Carnevale, 1997) and `Genesis` (see Bower & Beeman, 1998). These are available at `http://www.neuron.yale.edu` and `http://genesis.bbb.caltech.edu/GENESIS/genesis.html`.

response at these points. Specifically, the centroid at point x is defined as $\int dt\, t v(x,t) / \int dt\, v(x,t)$. Like the log-attenuation, the delay between any two points on a neuron is represented in the morphoelectrotonic transform as a distance.

Morphoelectrotonic transforms of a pyramidal cell from layer 5 of cat visual cortex are shown in figures 6.13 and 6.14. The left panel of figure 6.13 is a normal drawing of the neuron being studied, the middle panel shows the steady-state attenuation, and the right panel shows the delay. The transformed diagrams correspond to current being injected peripherally, with somatic potentials being compared to dendritic potentials. These figures indicate that, for potentials generated in the periphery, the apical and basal dendrites are much more uniform than the morphology would suggest.

The small neuron diagram at the upper left of figure 6.14 shows attenuation for the reverse situation from figure 6.13, when constant current is injected into the soma and dendritic potentials are compared with the somatic potential. Note how much smaller this diagram is than the one in the central panel of figure 6.13. This illustrates the general feature, mentioned previously, that potentials are attenuated much less in the outward than in the inward direction. This is because the thin dendrites provide less of a current sink for potentials arising from the soma than the soma provides for potentials coming from the dendrites.

The capacitance of neuronal cables causes the voltage attenuation for time-dependent current injection to increase as a function of frequency. Figure 6.14 compares the attenuation of dendritic potentials relative to the somatic potential when constant or sinusoidal current of two different frequencies is injected into the soma. Clearly, attenuation increases dramatically as a function of frequency. Thus, a neuron that appears electrotonically compact for static or low frequency current injection may be not compact when higher frequencies are considered. For example, action potential waveforms, which correspond to frequencies around 500 Hz, are much more severely attenuated within neurons than slower varying potentials.

6.4 Multi-compartment Models

The cable equation can be solved analytically only in relatively simple cases. When the complexities of real membrane conductances are included, the membrane potential must be computed numerically. This is done by splitting the modeled neuron into separate regions or compartments, and approximating the continuous membrane potential $V(x,t)$ by a discrete set of values representing the potentials within the different compartments. This assumes that each compartment is small enough so that there is negligible variation of the membrane potential across it. The precision of such a multi-compartmental description depends on the

where

$$B_\mu = c_m^{-1} g_{\mu,\mu-1}\,, \quad C_\mu = -c_m^{-1}\Big(\sum_i g_i^\mu + g_{\mu,\mu+1} + g_{\mu,\mu-1}\Big)\,,$$
$$D_\mu = c_m^{-1} g_{\mu,\mu+1}\,, \quad F_\mu = c_m^{-1}\Big(\sum_i g_i^\mu E_i + I_e^\mu / A_\mu\Big)\,. \tag{6.45}$$

Note that the gating variables and other parameters have been absorbed into the values of the coefficients B_μ, C_μ, D_μ, and F_μ in this equation. Equation 6.44, with μ running over all of the compartments of the model, generates a set of coupled differential equations. Because of the coupling between compartments, we cannot use the method discussed in appendix A of chapter 5 to integrate these equations. Instead, we present another method that shares some of the positive features of that approach. The Runge-Kutta method, which is a standard numerical integrator, is poorly suited for this application and is likely to run orders of magnitude slower than the method described below.

Two of the most important features of an integration method are accuracy and stability. Accuracy refers to how closely numerical finite-difference methods reproduce the exact solution of a differential equation as a function of the integration step size Δt. Stability refers to what happens when Δt is chosen to be excessively large and the method starts to become inaccurate. A stable integration method will degrade smoothly as Δt is increased, producing results of steadily decreasing accuracy. An unstable method, on the other hand, will at some point display a sudden transition and generate wildly inaccurate results. Given the tendency of impatient modelers to push the limits on Δt, it is highly desirable to have a method that is stable.

Defining

$$V_\mu(t + \Delta t) = V_\mu(t) + \Delta V_\mu\,, \tag{6.46}$$

the finite difference form of equation 6.44 gives the update rule

$$\Delta V_\mu = \big(B_\mu V_{\mu-1}(t) + C_\mu V_\mu(t) + D_\mu V_{\mu+1}(t) + F_\mu\big)\,\Delta t\,, \tag{6.47}$$

which is how ΔV_μ is computed using the so-called Euler method. This method is both inaccurate and unstable. The stability of the method can be improved dramatically by evaluating the membrane potentials on the right side of equation 6.47 not at time t, but at a later time $t + z\Delta t$, so that

$$\Delta V_\mu = \big(B_\mu V_{\mu-1}(t+z\Delta t) + C_\mu V_\mu(t+z\Delta t) + D_\mu V_{\mu+1}(t+z\Delta t) + F_\mu\big)\,\Delta t\,. \tag{6.48}$$

Two such methods are predominantly used, the reverse Euler method, for which $z = 1$, and the Crank-Nicholson method with $z = 0.5$. The reverse Euler method is the more stable of the two and the Crank-Nicholson is the more accurate. In either case, ΔV_μ is determined from equation 6.48. These methods are called implicit because equation 6.48 must be solved

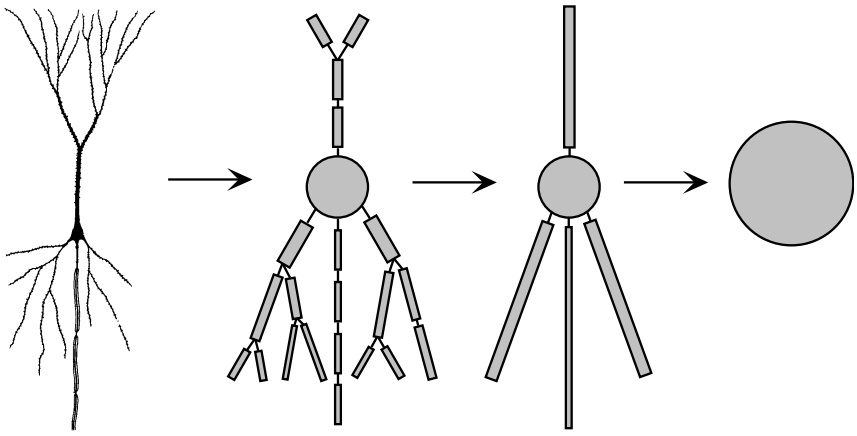

Figure 6.15 A sequence of approximations of the structure of a neuron. The neuron is represented by a variable number of discrete compartments, each representing a region that is described by a single membrane potential. The connectors between compartments represent resistive couplings. The simplest description is the single-compartment model furthest to the right.

three such terms, corresponding to coupling of the branching node to the first compartment in each of the daughter branches.

The constant $g_{\mu,\mu'}$ that determines the resistive coupling from neighboring compartment μ' to compartment μ is determined by computing the current that flows from one compartment to its neighbor due to Ohm's law. For simplicity, we begin by computing the coupling between two compartments that have the same length L and radius a. Using the results of chapter 5, the resistance between two such compartments, measured from their centers, is the intracellular resistivity, r_L times the distance between the compartment centers divided by the cross-sectional area, $r_L L/(\pi a^2)$. The total current flowing from compartment $\mu+1$ to compartment μ is then $\pi a^2 (V_{\mu+1} - V_\mu)/r_L L$. Equation 6.29 for the potential within a compartment μ refers to currents per unit area of membrane. Thus, we must divide the total current from compartment μ' by the surface area of compartment μ, $2\pi aL$, and we find that $g_{\mu,\mu'} = a/(2r_L L^2)$.

The value of $g_{\mu,\mu'}$ is given by a more complex expression if the two neighboring compartments have different lengths or radii. This can occur when a tapering cable is approximated by a sequence of cylindrical compartments, or at a branch point where a single compartment connects with two other compartments, as in figure 6.16. In either case, suppose that compartment μ has length L_μ and radius a_μ, and compartment μ' has length $L_{\mu'}$ and radius $a_{\mu'}$. The resistance between these two compartments is the sum of the two resistances from the middle of each compartment to the junction between them, $r_L L_\mu/(2\pi a_\mu^2) + r_L L_{\mu'}/(2\pi a_{\mu'}^2)$. To compute $g_{\mu,\mu'}$ we invert this expression and divide the result by the total surface area of

6.5 Chapter Summary

We continued the discussion of neuron modeling that began in chapter 5 by considering models with more complete sets of conductances and techniques for incorporating neuronal morphology. We introduced A-type K^+, transient Ca^{2+}, and Ca^{2+}-dependent K^+ conductances, and noted their effect on neuronal activity. The cable equation and its linearized version were introduced to examine the effects of morphology on membrane potentials. Finally, multi-compartment models were presented and used to discuss propagation of action potentials along unmyelinated and myelinated axons.

6.6 Appendices

A: Gating Functions for Conductance-Based Models

Connor-Stevens Model

The rate functions used for the gating variables n, m, and h of the Connor-Stevens model, in units of $1/\text{ms}$ with V in units of mV, are

$$\begin{aligned} \alpha_m &= \frac{0.38(V+29.7)}{1-\exp(-0.1(V+29.7))} & \beta_m &= 15.2\exp(-0.0556(V+54.7)) \\ \alpha_h &= 0.266\exp(-0.05(V+48)) & \beta_h &= 3.8/(1+\exp(-0.1(V+18))) \\ \alpha_n &= \frac{0.02(V+45.7)}{1-\exp(-0.1(V+45.7))} & \beta_n &= 0.25\exp(-0.0125(V+55.7))\,. \end{aligned} \tag{6.33}$$

The A-current is described directly in terms of the asymptotic values and τ functions for its gating variables (with τ_a and τ_b in units of ms and V in units of mV),

$$a_\infty = \left(\frac{0.0761\exp(0.0314(V+94.22))}{1+\exp(0.0346(V+1.17))}\right)^{1/3} \tag{6.34}$$

$$\tau_a = 0.3632 + 1.158/(1+\exp(0.0497(V+55.96))) \tag{6.35}$$

$$b_\infty = \left(\frac{1}{1+\exp(0.0688(V+53.3))}\right)^4 \tag{6.36}$$

and

$$\tau_b = 1.24 + 2.678/(1+\exp(0.0624(V+50)))\,. \tag{6.37}$$

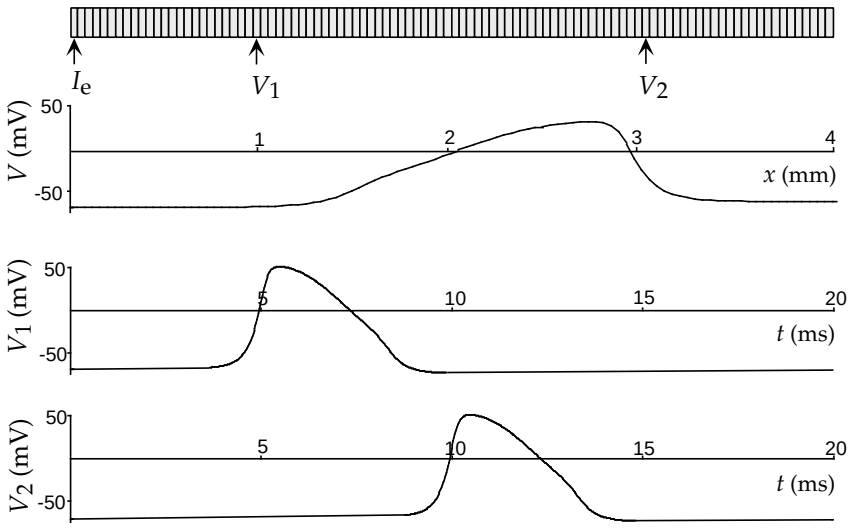

Figure 6.17 Propagation of an action potential along a multi-compartment model axon. The upper panel shows the multi-compartment representation of the axon with 100 compartments. The axon segment shown is 4 mm long and has a radius of 1 μm. An electrode current sufficient to initiate action potentials is injected at the point marked I_e. The panel beneath this shows the membrane potential as a function of position along the axon, at $t = 9.75$ ms. The spatial position in this panel is aligned with the axon depicted above it. The action potential is moving to the right. The bottom two panels show the membrane potential as a function of time at the two locations denoted by the arrows and symbols V_1 and V_2 in the upper panel.

representing a length of axon. Figure 6.17 shows an action-potential propagating along an axon modeled in this way. The action potential extends over more than 1 mm of axon and travels about 2 mm in 5 ms, for a speed of 0.4 m/s.

orthodromic and antidromic propagation

Although action potentials typically move along axons in a direction outward from the soma (called orthodromic propagation), the basic process of action-potential propagation does not favor one direction over the other. Propagation in the reverse direction, called antidromic propagation, is possible under certain stimulation conditions. For example, if an axon is stimulated in the middle of its length, action potentials will propagate in both directions away from the point of stimulation. Once an action potential starts moving along an axon, it does not generate a second action potential moving in the opposite direction because of refractory effects. The region in front of a moving action potential is ready to generate a spike as soon as enough current moves longitudinally down the axon from the region currently spiking to charge the next region up to spiking threshold. However, Na^+ conductances in the region just behind the moving action potential are still partially inactivated, so this region cannot generate another spike until after a recovery period. By the time the trailing region has recovered, the action potential has moved too far away to generate a second spike.